Question 1: **Incorrect**

Scenario: You have started at a new job at a company which has a Data Lake Storage Gen2 account. You have been tasked with uploading a single file to the account and you want to use a tool that you don't have to install or configure.

Which tool should you choose?

- ◉

  Azure Storage Explorer

  (Incorrect)

- ○

  Azure Data Studio

- ○

  Azure Data Catalogue

- ○

  The Azure Portal

  (Correct)

- ○

  Azure Data Factory

Explanation

The **Azure Portal** requires no installation or configuration. To upload a file, you only have to sign in and a select an Upload button.

https://docs.microsoft.com/en-us/azure/storage/files/storage-how-to-use-files-portal

Question 2: **Incorrect**

There are some core concepts with which data engineers should be familiar. The data engineer will often work with multiple types of data to perform many operations using many scripting or coding languages that are appropriate to their individual organization.

In an Azure Data Lake, data is stored in which of the following?

- ◉

A single JSON document

(Incorrect)

- ○

Files

(Correct)

- ○

None of the available options are correct

- ○

Relational tables

Explanation

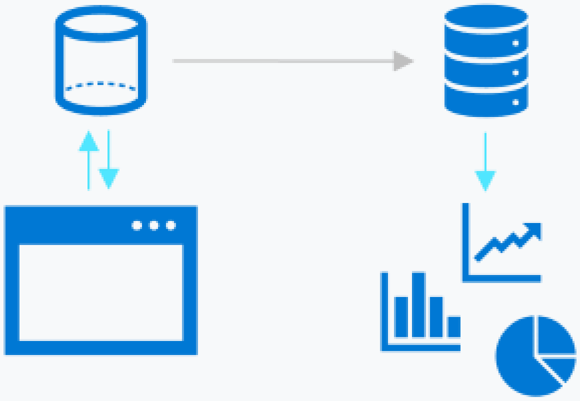
- *Data in a data lake is stored in files.*
- A data lake doesn't store data in relational tables.
- A data lake is based on a distributed file system, not a single JSON document.

Azure Data Lake works with existing IT investments for identity, management, and security for simplified data management and governance. It also integrates seamlessly with operational stores and data warehouses so you can extend current data applications.

Data engineering concepts

These concepts underpin many of the workloads that data engineers must implement and support.

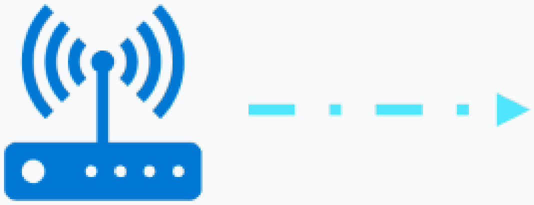
Operational and analytical data



Operational data is usually transactional data that is generated and stored by applications, often in a relational or non-relational database. *Analytical* data is data that has been optimized for analysis and reporting, often in a data warehouse.

One of the core responsibilities of a data engineer is to design, implement, and manage solutions that integrate operational and analytical data sources or extract operational data from multiple systems, transform it into appropriate structures for analytics, and load it into an analytical data store (usually referred to as ETL solutions).

Streaming data



Streaming data refers to perpetual sources of data that generate data values in real-time, often relating to specific events. Common sources of streaming data include internet-of-things (IoT) devices and social media feeds.

Data engineers often need to implement solutions that capture real-time stream of data and ingest them into analytical data systems, often combining the real-time data with other application data that is processed in batches.

Data pipelines

Data pipelines are used to orchestrate activities that transfer and transform data. Pipelines are the primary way in which data engineers implement repeatable extract, transform, and load (ETL) solutions that can be triggered based on a schedule or in response to events.

Data lakes

A data lake is a storage repository that holds large amounts of data in native, raw formats. Data lake stores are optimized for scaling to massive volumes (terabytes or 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.

Data warehouses

A data warehouse is a centralized repository of integrated data from one or more disparate sources. Data warehouses store current and historical data in relational tables that are organized into a schema that optimizes performance for analytical queries.

Data engineers are responsible for designing and implementing relational data warehouses, and managing regular data loads into tables.

Apache Spark

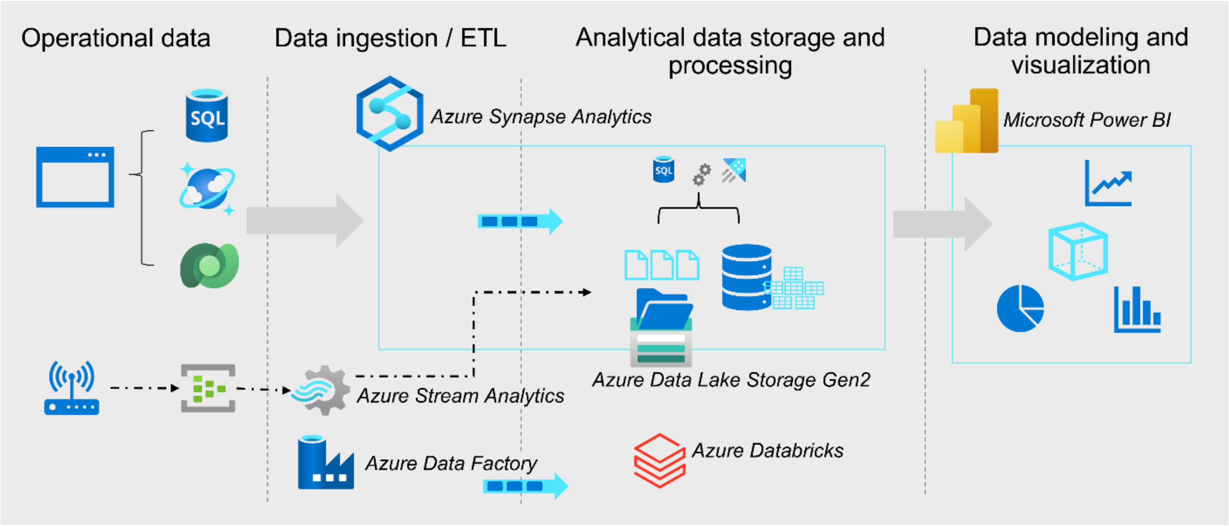
Apache Spark is a parallel processing framework that takes advantage of in-memory processing and a distributed file storage. It's a common open-source software (OSS) tool for big data scenarios.

Data engineers need to be proficient with Spark, using notebooks and other code artifacts to process data in a data lake and prepare it for modelling and analysis.

https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-a-data-lake/#what-is-a-data-lake

Question 3: **Incorrect**

Microsoft Azure includes many services that can be used to implement and manage data engineering workloads.



Which of the following Azure services provides capabilities for running data pipelines AND managing analytical data in a data lake or relational data warehouse?

- ●

  Azure Databricks

  (Incorrect)
- ○

  Azure Data Explorer
- ○

  Azure Stream Analytics
- ○

  Azure Synapse Analytics

  (Correct)

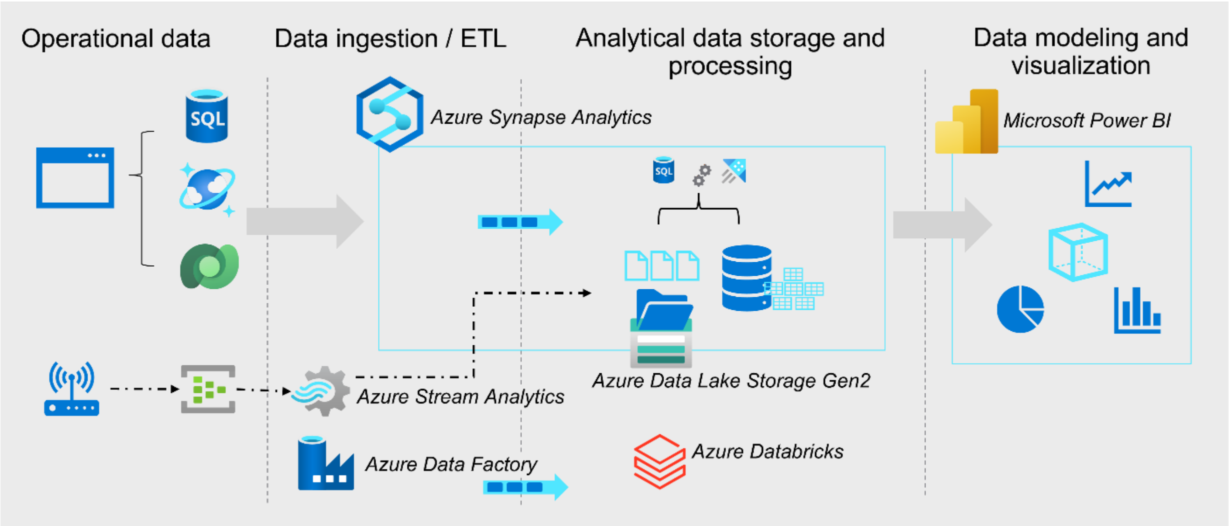
Explanation

- *Azure Synapse Analytics includes functionality for pipelines, data lakes, and relational data warehouses.*
- Azure Stream Analytics is used to process real-time streams of data.

- Azure Databricks is primarily an Apache Spark implementation for processing data in a data lake.
- Azure Synapse Data Explorer provides customers with an interactive query experience to unlock insights from log and telemetry data. To complement existing SQL and Apache Spark analytics runtime engines, Data Explorer analytics runtime is optimized for efficient log analytics using powerful indexing technology to automatically index free-text and semi-structured data commonly found in the telemetry data.

Data engineering in Microsoft Azure

Microsoft Azure includes many services that can be used to implement and manage data engineering workloads.



The diagram displays the flow from left to right of a typical enterprise data analytics solution, including some of the key Azure services that may be used. Operational data is generated by applications and devices and stored in Azure data storage services such as Azure SQL Database, Azure Cosmos DB, and Microsoft Dataverse. Streaming data is captured in event broker services such as Azure Event Hubs.

This operational data must be captured, ingested, and consolidated into analytical stores; from where it can be modelled and visualized in reports and dashboards. These tasks represent the core area of responsibility for the data engineer. The core Azure technologies used to implement data engineering workloads include:

- Azure Synapse Analytics

- Azure Data Lake Storage Gen2
- Azure Stream Analytics
- Azure Data Factory
- Azure Databricks

The analytical data stores that are populated with data produced by data engineering workloads support data modelling and visualization for reporting and analysis, often using sophisticated visualization tools such as Microsoft Power BI.

https://learn.microsoft.com/en-us/azure/synapse-analytics/overview-what-is

Question 4: **Correct**

The SQL language includes many features and functions that enable you to manipulate data. For example, you can use SQL to:

- Filter rows and columns in a dataset.
- Rename data fields and convert between data types.
- Calculate derived data fields.
- Manipulate string values.
- Group and aggregate data.

Azure Synapse serverless SQL pools can be used to run SQL statements that transform data and persist the results as a file in a data lake for further processing or querying.

Which external database object encapsulates the connection information to a file location in a data lake store?

- ●

  `DATA SOURCE`

  (Correct)

- ○

  `ROWSET`

- ○

  `OPENROWSET`

- ○

  `FILE FORMAT`

- ○

  `EXTERNAL TABLE`

Explanation

A database is an object held within a serverless SQL pool to hold data and metadata objects.

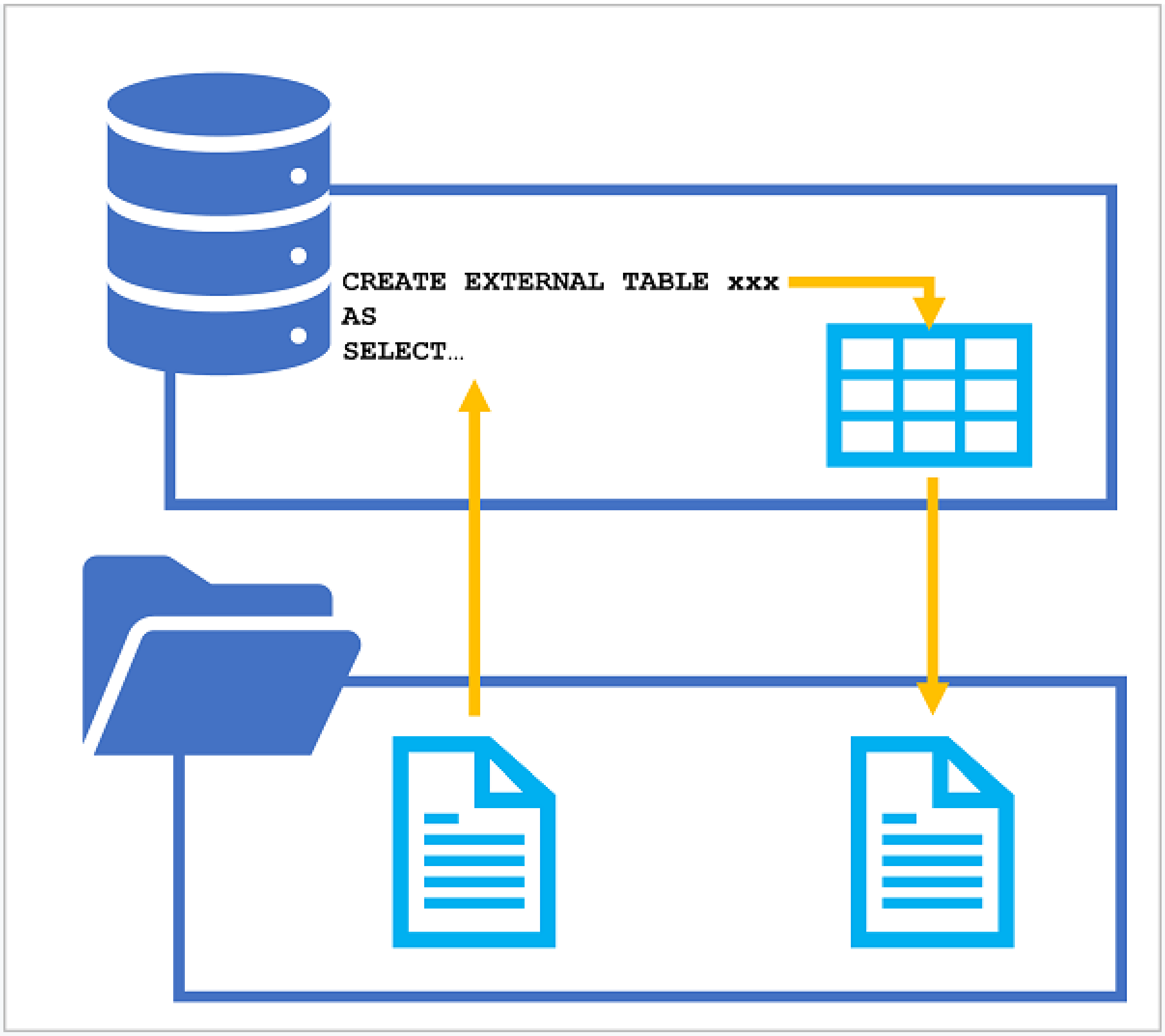
- *A `DATA SOURCE` provides the connection information to the files in a data lake store.*
- An `EXTERNAL TABLE` creates the table object without selecting data into it.
- A `FILE FORMAT` is the structure of a file that tells a program how to display its contents.
- The `OPENROWSET` is used to read the data in files stored in a data lake.
- The `ROWSET` is not a valid function.

Transform data files with the `CREATE EXTERNAL TABLE AS SELECT` statement

Azure Synapse serverless SQL pools can be used to run SQL statements that transform data and persist the results as a file in a data lake for further processing or querying. If you're familiar with Transact-SQL syntax, you can craft a `SELECT` statement that applies the specific transformation you're interested in, and store the results of the `SELECT` statement in a selected file format with a metadata table schema that can be queried using SQL.

You can use a `CREATE EXTERNAL TABLE AS SELECT` (CETAS) statement in a dedicated SQL pool or serverless SQL pool to persist the results of a query in an external table, which stores its data in a file in the data lake.

The CETAS statement includes a `SELECT` statement that queries and manipulates data from any valid data source (which could be an existing table or view in a database, or an `OPENROWSET` function that reads file-based data from the data lake). The results of the `SELECT` statement are then persisted in an external table, which is a metadata object in a database that provides a relational abstraction over data stored in files. The following diagram illustrates the concept visually:



By applying this technique, you can use SQL to extract and transform data from files or tables, and store the transformed results for downstream processing or analysis. Subsequent operations on the transformed data can be performed against the relational table in the SQL pool database or directly against the underlying data files.

Creating external database objects to support CETAS

To use CETAS expressions, you must create the following types of object in a database for either a serverless or dedicated SQL pool. When using a serverless SQL pool, create these objects in a custom database (created using the `CREATE DATABASE` statement), not the **built-in** database.

External data source

An external data source encapsulates a connection to a file system location in a data lake. You can then use this connection to specify a relative path in which the data files for the external table created by the CETAS statement are saved.

If the source data for the CETAS statement is in files in the same data lake path, you can use the same external data source in the `OPENROWSET` function used to query it. Alternatively, you can create a separate external data source for the source files or use a fully qualified file path in the `OPENROWSET` function.

To create an external data source, use the `CREATE EXTERNAL DATA SOURCE` statement, as shown in this example:

```
SQL
CREATE EXTERNAL DATA SOURCE files
WITH (
    LOCATION = 'https://mydatalake.blob.core.windows.net/data/files/'
);
```

The previous example assumes that users running queries that use the external data source will have sufficient permissions to access the files. An alternative approach is to encapsulate a credential in the external data source so that it can be used to access file data without granting all users permissions to read it directly:

```
SQL
CREATE DATABASE SCOPED CREDENTIAL storagekeycred
WITH
    IDENTITY='SHARED ACCESS SIGNATURE',
    SECRET = 'sv=xxx...';

CREATE EXTERNAL DATA SOURCE secureFiles
WITH (
    LOCATION = 'https://mydatalake.blob.core.windows.net/data/secureFiles/'
    CREDENTIAL = storagekeycred
);
```

In addition to SAS authentication, you can define credentials that use managed identity (the Azure Active Directory identity used by your Azure Synapse workspace), a specific

Azure Active Directory principal, or passthrough authentication based on the identity of the user running the query (which is the default type of authentication).

https://learn.microsoft.com/en-us/azure/synapse-analytics/sql/develop-storage-files-storage-access-control

External file format

The CETAS statement creates a table with its data stored in files. You must specify the format of the files you want to create as an external file format.

To create an external file format, use the `CREATE EXTERNAL FILE FORMAT` statement, as shown in this example:

```
SQL
CREATE EXTERNAL FILE FORMAT ParquetFormat
WITH (
        FORMAT_TYPE = PARQUET,
        DATA_COMPRESSION = 'org.apache.hadoop.io.compress.SnappyCodec'
    );
```

In this example, the files will be saved in Parquet format. You can also create external file formats for other types of file.

https://learn.microsoft.com/en-us/sql/t-sql/statements/create-external-file-format-transact-sql

Using the CETAS statement

After creating an external data source and external file format, you can use the CETAS statement to transform data and stored the results in an external table.

For example, suppose the source data you want to transform consists of sales orders in comma-delimited text files that are stored in a folder in a data lake. You want to filter the data to include only orders that are marked as "special order", and save the transformed data as Parquet files in a different folder in the same data lake. You could use the same external data source for both the source and destination folders as shown in this example:

```
SQL
CREATE EXTERNAL TABLE SpecialOrders
    WITH (
        -- details for storing results
        LOCATION = 'special_orders/',
        DATA_SOURCE = files,
        FILE_FORMAT = ParquetFormat
    )
AS
SELECT OrderID, CustomerName, OrderTotal
FROM
    OPENROWSET(
        -- details for reading source files
        BULK 'sales_orders/*.csv',
        DATA_SOURCE = 'files',
        FORMAT = 'CSV',
        PARSER_VERSION = '2.0',
        HEADER_ROW = TRUE
    ) AS source_data
WHERE OrderType = 'Special Order';
```

The `LOCATION` and `BULK` parameters in the previous example are relative paths for the results and source files respectively. The paths are relative to the file system location referenced by the **files** external data source.

An important point to understand is that you ***must*** use an external data source to specify the location where the transformed data for the external table is to be saved. When file-based source data is stored in the same folder hierarchy, you can use the same external data source. Otherwise, you can use a second data source to define a connection to the source data or use the fully qualified path, as shown in this example:

```
SQL
CREATE EXTERNAL TABLE SpecialOrders
    WITH (
        -- details for storing results
        LOCATION = 'special_orders/',
        DATA_SOURCE = files,
        FILE_FORMAT = ParquetFormat
    )
AS
SELECT OrderID, CustomerName, OrderTotal
FROM
    OPENROWSET(
        -- details for reading source files
        BULK
'https://mystorage.blob.core.windows.net/data/sales_orders/*.csv',
        FORMAT = 'CSV',
        PARSER_VERSION = '2.0',
        HEADER_ROW = TRUE
    ) AS source_data
WHERE OrderType = 'Special Order';
```

Dropping external tables

If you no longer need the external table containing the transformed data, you can drop it from the database my using the `DROP EXTERNAL TABLE` statement, as shown here:

```
SQL
DROP EXTERNAL TABLE SpecialOrders;
```

However, it's important to understand that external tables are a metadata abstraction over the files that contain the actual data. Dropping an external table does ***not*** delete the underlying files.

https://learn.microsoft.com/en-us/azure/synapse-analytics/sql/develop-tables-cetas

Question 5: **Correct**

True or False: Materialized views are prewritten queries with joins and filters whose definition is saved and the results persisted to both serverless and dedicated SQL pools.

- ●

  False

  (Correct)

- ○

  True

Explanation

Materialized views are prewritten queries with joins and filters whose definition is saved and the results persisted to a dedicated SQL pool. They are not supported by serverless SQL pools.

Materialized views results in increased performance since the data within the view can be fetched without having to resolve the underlying query to base tables. You can also further filter and supplement other queries as if it is a table also. In addition, you also can define a different table distribution within the materialized view definition that is different from the table on which it is based.

As a result, you can use Materialized Views to improve the performance of either complex or slow queries. As the data in the underlying base tables change, the data in the materialized view will automatically update without user interaction.

There are several restrictions that you must be aware of before defining a materialized view:

• The `SELECT` list in the materialized view definition needs to meet at least one of these two criteria:

• The `SELECT` list contains an aggregate function.

• `GROUP BY` is used in the Materialized view definition and all columns in `GROUP BY` are included in the `SELECT` list. Up to 32 columns can be used in the `GROUP BY` clause.

• Supported aggregations include `MAX`, `MIN`, `AVG`, `COUNT`, `COUNT_BIG`, `SUM`, `VAR`, `STDEV`.

• Only the hash and round_robin table distribution is supported in the definition.

• Only `CLUSTERED COLUMNSTORE INDEX` is supported by materialized view.

The following is an example of creating a materialized view named myview, using a hash distribution selecting two columns from a table and grouping by them.

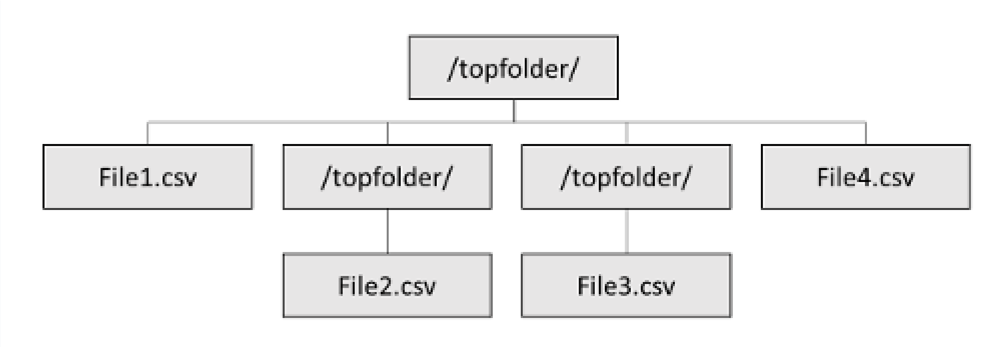
```
SQL
create materialized view mview
with(distribution=hash(col1))
as select col1, col2 from dbo.table group by col1, col2;
```

https://docs.microsoft.com/en-us/sql/t-sql/statements/create-materialized-view-as-select-transact-sql?view=azure-sqldw-latest

Question 6: **Incorrect**

Scenario: Dr. Karl Malus works for the Power Broker Corporation founded by Curtiss Jackson, using technology to service various countries and their military efforts. You have been contracted by the company to assist Dr. Malus with some Azure Data Lake Storage work.

Dr. Malus has files and folders in Azure Data Lake Storage Gen2 for an Azure Synapse workspace as shown in the following exhibit.



A team member creates an external table named ExtTable that has `LOCATION='/topfolder/'`.

When Dr. Malus queries ExtTable by using an Azure Synapse Analytics serverless SQL pool, which of the following files are returned? (Select all that apply)

- ☑

  File3.csv

  (Incorrect)

- □

  File4.csv

  (Correct)

- □

  File2.csv

- □

  File1.csv

  (Correct)

Explanation
Serverless SQL pool can recursively traverse folders only if you specify `/**` *at the end of path.*

Serverless SQL pool supports reading multiple files/folders by using wildcards, which are similar to the wildcards used in Windows OS. However, greater flexibility is present since multiple wildcards are allowed.

https://docs.microsoft.com/en-us/azure/synapse-analytics/sql/query-folders-multiple-csv-files

In case of a serverless pool a wildcard should be added to the location.

When reading from Parquet files, you can specify only the columns you want to read and skip the rest.

`LOCATION = 'folder_or_filepath'`

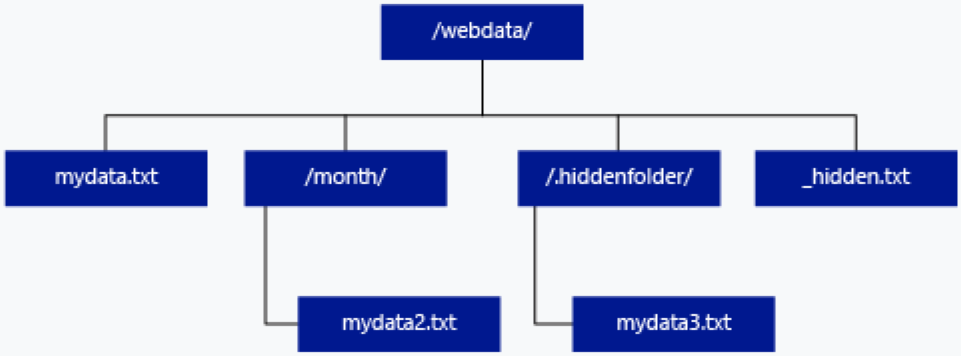
Specifies the folder or the file path and file name for the actual data in Azure Blob Storage. The location starts from the root folder. The root folder is the data location specified in the external data source.

Unlike Hadoop external tables, native external tables don't return subfolders unless you specify `/**` at the end of path. In this example, if `LOCATION='/webdata/'`, a serverless SQL pool query, will return rows from `mydata.txt`. It won't return `mydata2.txt` and `mydata3.txt` because they're located in a subfolder. Hadoop tables will return all files within any sub-folder.

Both Hadoop and native external tables will skip the files with the names that begin with an underline (_) or a period (.).

`DATA_SOURCE = external_data_source_name` - Specifies the name of the external data source that contains the location of the external data. To create an external data source, use CREATE EXTERNAL DATA SOURCE.

`FILE_FORMAT = external_file_format_name` - Specifies the name of the external file format object that stores the file type and compression method for the external data.



https://docs.microsoft.com/en-us/azure/synapse-analytics/sql/develop-tables-external-tables?tabs=hadoop#arguments-create-external-table

Question 7: **Correct**
You can use a certain function in SQL queries that run in the default master database of the built-in serverless SQL pool to explore data in the data lake. However, sometimes you may want to create a custom database that contains some objects that make it easier to work with external data in the data lake that you need to query frequently.

Which of the following functions is used to read the data in files stored in a data lake?

- ●

  `OPENROWSET`

  (Correct)

- ○

  `FORMAT`

- ○

  `BULK`

- ○

  None of the provided options are correct

- ○

  `ROWSET`

Explanation

- *The `OPENROWSET` is used to read the data in files stored in a data lake.*
- The `FORMAT` is a clause within the `OPENROWSET` function to determine the file type to be read.
- The `ROWSET` is not a valid function.
- The `BULK` parameter is used to specify the relative path for certain file types in a specified folder in the below example.

Create external database objects

You can use the `OPENROWSET` function in SQL queries that run in the default **master** database of the built-in serverless SQL pool to explore data in the data lake. However, sometimes you may want to create a custom database that contains some objects that make it easier to work with external data in the data lake that you need to query frequently.

Creating a database

You can create a database in a serverless SQL pool just as you would in a SQL Server instance. You can use the graphical interface in Synapse Studio, or a `CREATE DATABASE` statement. One consideration is to set the collation of your database so that it supports conversion of text data in files to appropriate Transact-SQL data types.

The following example code creates a database named *salesDB* with a collation that makes it easier to import UTF-8 encoded text data into `VARCHAR` columns.

```
SQL
CREATE DATABASE SalesDB
    COLLATE Latin1_General_100_BIN2_UTF8
```

Creating an external data source

You can use the `OPENROWSET` function with a BULK path to query file data from your own database, just as you can in the **master** database; but if you plan to query data in the same location frequently, it's more efficient to define an external data source that references that location. For example, the following code creates a data source named *files* for the hypothetical `https://mydatalake.blob.core.windows.net/data/files/` folder:

```
SQL
CREATE EXTERNAL DATA SOURCE files
WITH (
    LOCATION = 'https://mydatalake.blob.core.windows.net/data/files/'
)
```

One benefit of an external data source, is that you can simplify an `OPENROWSET` query to use the combination of the data source and the relative path to the folders or files you want to query:

```
SQL
SELECT *
FROM
    OPENROWSET(
        BULK 'orders/*.csv',
        DATA_SOURCE = 'files',
        FORMAT = 'csv',
        PARSER_VERSION = '2.0'
    ) AS orders
```

In this example, the `BULK` parameter is used to specify the relative path for all .csv files in the **orders** folder, which is a subfolder of the **files** folder referenced by the data source.

Another benefit of using a data source is that you can assign a credential for the data source to use when accessing the underlying storage, enabling you to provide access to data through SQL without permitting users to access the data directly in the storage account. For example, the following code creates a credential that uses a shared access signature (SAS) to authenticate against the underlying Azure storage account hosting the data lake.

```
SQL
CREATE DATABASE SCOPED CREDENTIAL sqlcred
WITH
    IDENTITY='SHARED ACCESS SIGNATURE',
    SECRET = 'sv=xxx...';
GO

CREATE EXTERNAL DATA SOURCE secureFiles
WITH (
    LOCATION = 'https://mydatalake.blob.core.windows.net/data/secureFiles/'
    CREDENTIAL = sqlcred
);
GO
```

In addition to SAS authentication, you can define credentials that use managed identity (the Azure Active Directory identity used by your Azure Synapse workspace), a specific Azure Active Directory principal, or passthrough authentication based on the identity of the user running the query (which is the default type of authentication).

https://learn.microsoft.com/en-us/azure/synapse-analytics/sql/develop-storage-files-storage-access-control

Creating an external file format

While an external data source simplifies the code needed to access files with the OPENROWSET function, you still need to provide format details for the file being access; which may include multiple settings for delimited text files. You can encapsulate these settings in an external file format, like this:

```
SQL
CREATE EXTERNAL FILE FORMAT CsvFormat
    WITH (
        FORMAT_TYPE = DELIMITEDTEXT,
        FORMAT_OPTIONS(
            FIELD_TERMINATOR = ',',
            STRING_DELIMITER = '"'
        )
    );
GO
```

After creating file formats for the specific data files you need to work with, you can use the file format to create external tables, as discussed next.

Creating an external table

When you need to perform a lot of analysis or reporting from files in the data lake, using the OPENROWSET function can result in complex code that includes data sources and file paths. To simplify access to the data, you can encapsulate the files in an external table; which users and reporting applications can query using a standard SQL SELECT statement just like any other database table. To create an external table, use the CREATE EXTERNAL TABLE statement, specifying the column schema as for a standard table, and including a WITH clause specifying the external data source, relative path, and external file format for your data.

```
SQL
CREATE EXTERNAL TABLE dbo.products
(
    product_id INT,
    product_name VARCHAR(20),
    list_price DECIMAL(5,2)
)
```

```
WITH
(
    DATA_SOURCE = files,
    LOCATION = 'products/*.csv',
    FILE_FORMAT = CsvFormat
);
GO

-- query the table
SELECT * FROM dbo.products;
```

By creating a database that contains the external objects discussed in this unit, you can provide a relational database layer over files in a data lake, making it easier for many data analysts and reporting tools to access the data by using standard SQL query semantics.

https://learn.microsoft.com/en-us/sql/t-sql/functions/openrowset-transact-sql?view=sql-server-ver16

Question 8: **Correct**

What are Azure Synapse Studio notebooks based on?

- ◉

  Spark

  (Correct)

- ○

  Spark Pool

- ○

  SQL Pool

- ○

  T-SQL

Explanation

Azure Synapse Studio notebook is purely Spark based.

https://docs.microsoft.com/en-us/azure/synapse-analytics/spark/apache-spark-development-using-notebooks?tabs=classical

Question 9: **Incorrect**
How do you cache data into the memory of the local executor for instant access?

- ●

  `.inMemory().save()`

  (Incorrect)

- ○

  `.cacheLocalExe()`

- ○

  `.save().inMemory()`

- ○

  `.cache()`

  (Correct)

Explanation
The `.cache()` method is an alias for `.persist()`. Calling this moves data into the memory of the local executor.

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

Question 10: **Incorrect**
You can use a serverless SQL pool to query data files in various common file formats, including:

- Delimited text, such as comma-separated values (CSV) files.
- JavaScript object notation (JSON) files.
- Parquet files.

The basic syntax for querying is the same for all of these types of file, and is built on the `OPENROWSET` SQL function; which generates a tabular rowset from data in one or more files.

Which character in file path can be used to select all the file/folders that match rest of the path?

- ●

//

(Incorrect)

- ○

&

- ○

*****

(Correct)

- ○

/

Explanation

- *The asterisk character ' * ' in file path can be used to select all the file or folders that match rest of the path. See example below.*
- The ampersand character ' & ' is used to concatenate fields in the files together.
- The forward slash ' / ' is used as a directory separator in folder and file path.
- The double forward slash is generally used to denote a comment or explanation that should be ignored by the compiler or computer. It is also part of a URL such as in https://www.microsoft.com

https://stackoverflow.com/questions/40056213/behavior-of-asterisk-in-azure-search-service

Query files using a serverless SQL pool

You can use a serverless SQL pool to query data files in various common file formats, including:

- Delimited text, such as comma-separated values (CSV) files.
- JavaScript object notation (JSON) files.
- Parquet files.

The basic syntax for querying is the same for all of these types of file, and is built on the `OPENROWSET` SQL function; which generates a tabular rowset from data in one or more files. For example, the following query could be used to extract data from CSV files.

```
SQL
SELECT TOP 100 *
FROM OPENROWSET(
    BULK 'https://mydatalake.blob.core.windows.net/data/files/*.csv',
    FORMAT = 'csv') AS rows
```

The `OPENROWSET` function includes more parameters that determine factors such as:

- The schema of the resulting rowset
- Additional formatting options for delimited text files.

https://learn.microsoft.com/en-us/azure/synapse-analytics/sql/develop-openrowset#syntax.

The output from `OPENROWSET` is a rowset to which an alias must be assigned. In the previous example, the alias **rows** is used to name the resulting rowset.

The `BULK` parameter includes the full URL to the location in the data lake containing the data files. This can be an individual file, or a folder with a wildcard expression to filter the file types that should be included. The `FORMAT` parameter specifies the type of data being queried. The example above reads delimited text from all .csv files in the **files** folder.

https://learn.microsoft.com/en-us/azure/synapse-analytics/sql/develop-openrowset#syntax

Note: This example assumes that the user has access to the files in the underlying store, If the files are protected with a SAS key or custom identity, you would need to create a server-scoped credential.

As seen in the previous example, you can use wildcards in the `BULK` parameter to include or exclude files in the query. The following list shows a few examples of how this can be used:

- `https://mydatalake.blob.core.windows.net/data/files/file1.csv`: Only include *file1.csv* in the *files* folder.

- `https://mydatalake.blob.core.windows.net/data/files/file*.csv`: All .csv files in the *files* folder with names that start with "file".
- `https://mydatalake.blob.core.windows.net/data/files/*`: All files in the *files* folder.
- `https://mydatalake.blob.core.windows.net/data/files/**`: All files in the *files* folder, and recursively its subfolders.

You can also specify multiple file paths in the `BULK` parameter, separating each path with a comma.

Querying delimited text files

Delimited text files are a common file format within many businesses. The specific formatting used in delimited files can vary, for example:

- With and without a header row.
- Comma and tab-delimited values.
- Windows and Unix style line endings.
- Non-quoted and quoted values, and escaping characters.

Regardless of the type of delimited file you're using, you can read data from them by using the `OPENROWSET` function with the **csv** FORMAT parameter, and other parameters as required to handle the specific formatting details for your data. For example:

```
SQL
SELECT TOP 100 *
FROM OPENROWSET(
    BULK 'https://mydatalake.blob.core.windows.net/data/files/*.csv',
    FORMAT = 'csv',
    PARSER_VERSION = '2.0',
    FIRSTROW = 2) AS rows
```

The `PARSER_VERSION` is used to determine how the query interprets the text encoding used in the files. Version 1.0 is the default and supports a wide range of file encodings, while version 2.0 supports fewer encodings but offers better performance.
The `FIRSTROW` parameter is used to skip rows in the text file, to eliminate any unstructured preamble text or to ignore a row containing column headings.

Additional parameters you might require when working with delimited text files include:

- `FIELDTERMINATOR` - the character used to separate field values in each row. For example, a tab-delimited file separates fields with a TAB (*\t*) character. The default field terminator is a comma (*,*).
- `ROWTERMINATOR` - the character used to signify the end of a row of data. For example, a standard Windows text file uses a combination of a carriage return (CR) and line feed (LF), which is indicated by the code *\n*; while UNIX-style text files use a single line feed character, which can be indicated using the code *0x0a*.
- `FIELDQUOTE` - the character used to enclose quoted string values. For example, to ensure that the comma in the address field value *126 Main St, apt 2* isn't interpreted as a field delimiter, you might enclose the entire field value in quotation marks like this: *"126 Main St, apt 2"*. The double-quote (") is the default field quote character.

Specifying the rowset schema

It's common for delimited text files to include the column names in the first row. The OPENROWSET function can use this to define the schema for the resulting rowset, and automatically infer the data types of the columns based on the values they contain. For example, consider the following delimited text:

```
text
product_id,product_name,list_price
123,Widget,12.99
124,Gadget,3.99
```

The data consists of the following three columns:

- **product_id** (integer number)
- **product_name** (string)
- **list_price** (decimal number)

You could use the following query to extract the data with the correct column names and appropriately inferred SQL Server data types (in this case `INT`, `NVARCHAR`, and `DECIMAL`)

```
SQL
SELECT TOP 100 *
FROM OPENROWSET(
    BULK 'https://mydatalake.blob.core.windows.net/data/files/*.csv',
    FORMAT = 'csv',
    PARSER_VERSION = '2.0',
    HEADER_ROW = TRUE) AS rows
```

Question 11: **Incorrect**

Identify the missing word(s) in the following sentence within the context of Microsoft Azure.

[?] describes the final cost of owning a given technology. In on-premises systems, [?] includes the following costs:

- Hardware

- Software licensing

- Labour (installation, upgrades, maintenance)

- Datacentre overhead (power, telecommunications, building, heating and cooling)

- ●

  MTD

  (Incorrect)

- ○

  TCO

  (Correct)

- ○

  RPO

- ○

  RTO

Explanation

Total cost of ownership

The term *total cost of ownership* (TCO) describes the final cost of owning a given technology. In on-premises systems, TCO includes the following costs:

- Hardware

• Software licensing

• Labour (installation, upgrades, maintenance)

• Datacentre overhead (power, telecommunications, building, heating and cooling)

It's difficult to align on-premises expenses with actual usage. Organizations buy servers that have extra capacity so they can accommodate future growth. A newly purchased server will always have excess capacity that isn't used. When an on-premises server is at maximum capacity, even an incremental increase in resource demand will require the purchase of more hardware.

Because on-premises server systems are very expensive, costs are often *capitalized*. This means that on financial statements, costs are spread out across the expected lifetime of the server equipment. Capitalization restricts an IT manager's ability to buy upgraded server equipment during the expected lifetime of a server. This restriction limits the server system's ability to accommodate increased demand.

In cloud solutions, expenses are recorded on the financial statements each month. They're monthly expenses instead of capital expenses. Because subscriptions are a different kind of expense, the expected server lifetime doesn't limit the IT manager's ability to upgrade to meet an increase in demand.

https://www.purchasing-procurement-center.com/total-cost-of-ownership.html

Question 12: **Correct**
Consider: Azure Data Factory diagnostic logs

By default, how long are the Azure Data Factory diagnostic logs retained for?

- ◉

  None of the listed options.

  (Correct)

- ○

  40 days

- ○

  20 days

- ○

  50 days

- ○

  30 days

- ○

  10 days

Explanation

Data Factory stores pipeline-run data for only 45 days. Use Azure Monitor if you want to keep that data for a longer time. With Monitor, you can route diagnostic logs for analysis to multiple different targets.

• **Storage Account**: Save your diagnostic logs to a storage account for auditing or manual inspection. You can use the diagnostic settings to specify the retention time in days.

• **Event Hub**: Stream the logs to Azure Event Hubs. The logs become input to a partner service/custom analytics solution like Power BI.

• **Log Analytics**: Analyze the logs with Log Analytics. The Data Factory integration with Azure Monitor is useful in the following scenarios:

• You want to write complex queries on a rich set of metrics that are published by Data Factory to Monitor. You can create custom alerts on these queries via Monitor.

• You want to monitor across data factories. You can route data from multiple data factories to a single Monitor workspace.

https://docs.microsoft.com/en-us/azure/data-factory/monitor-using-azure-monitor

Question 13: **Incorrect**

Identify the missing word(s) in the following sentence within the context of Microsoft Azure.

[?] data is typically stored in a relational database such as SQL Server or Azure SQL Database.

- ●

  Semi-structured

(Incorrect)

- ○

  Unstructured

- ○

  Structured

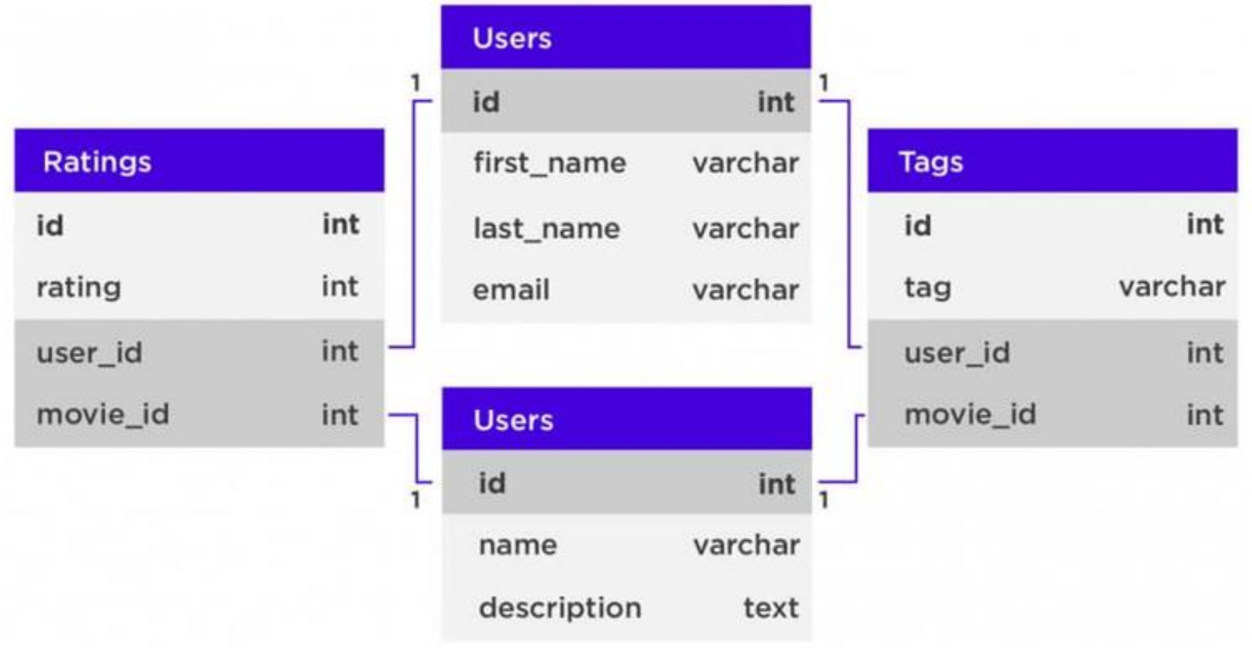
  (Correct)

- ○

  JSON Format

Explanation

Depending on the type of data such as structured, semi-structured, or unstructured, data will be stored differently. Structured data is typically stored in a relational database such as SQL Server or Azure SQL Database. Azure SQL Database is a service that runs in the cloud.



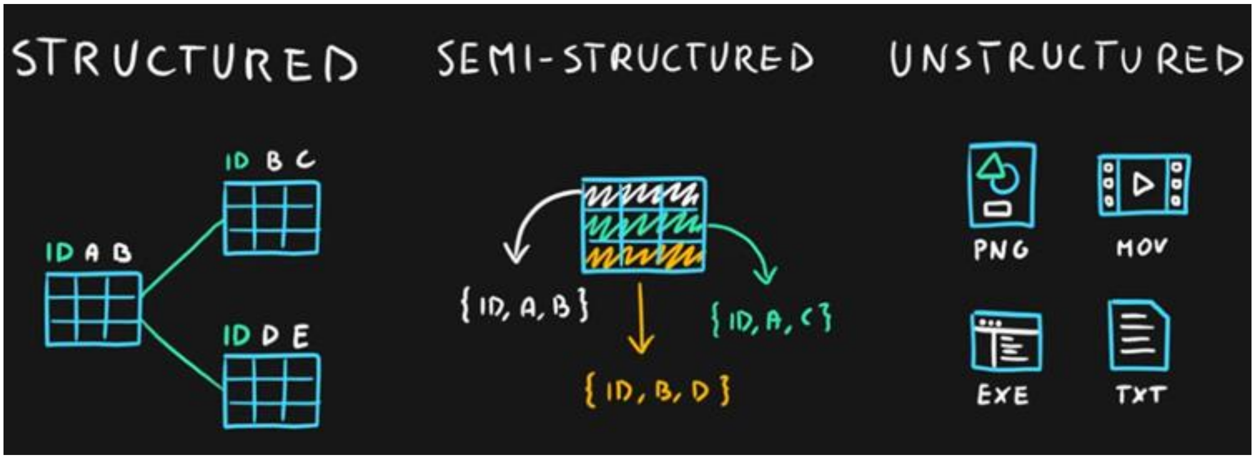
Ratings	
id	int
rating	int
user_id	int
movie_id	int

Users	
id	int
first_name	varchar
last_name	varchar
email	varchar

Tags	
id	int
tag	varchar
user_id	int
movie_id	int

Users	
id	int
name	varchar
description	text

You can use it to create and access relational tables. The service is managed and run by Azure, you just specify that you want a database server to be created. The act of setting up the database server is called *provisioning*.



https://f5a395285c.nxcli.net/microsoft-azure/dp-900/structured-data-vs-unstructured-data-vs-semi-structured-data/

Question 14: **Incorrect**
Azure Advisor provides you with personalized messages that provide information on best practices to optimize the setup of your Azure services. It analyzes your resource configuration and usage telemetry and then recommends solutions for which of the following Azure metrics? (Select all that apply)

- ☑

  Security

  (Correct)

- □

  Cost effectiveness

  (Correct)

- □

  Performance

(Correct)

- □

  Encryption deficiencies

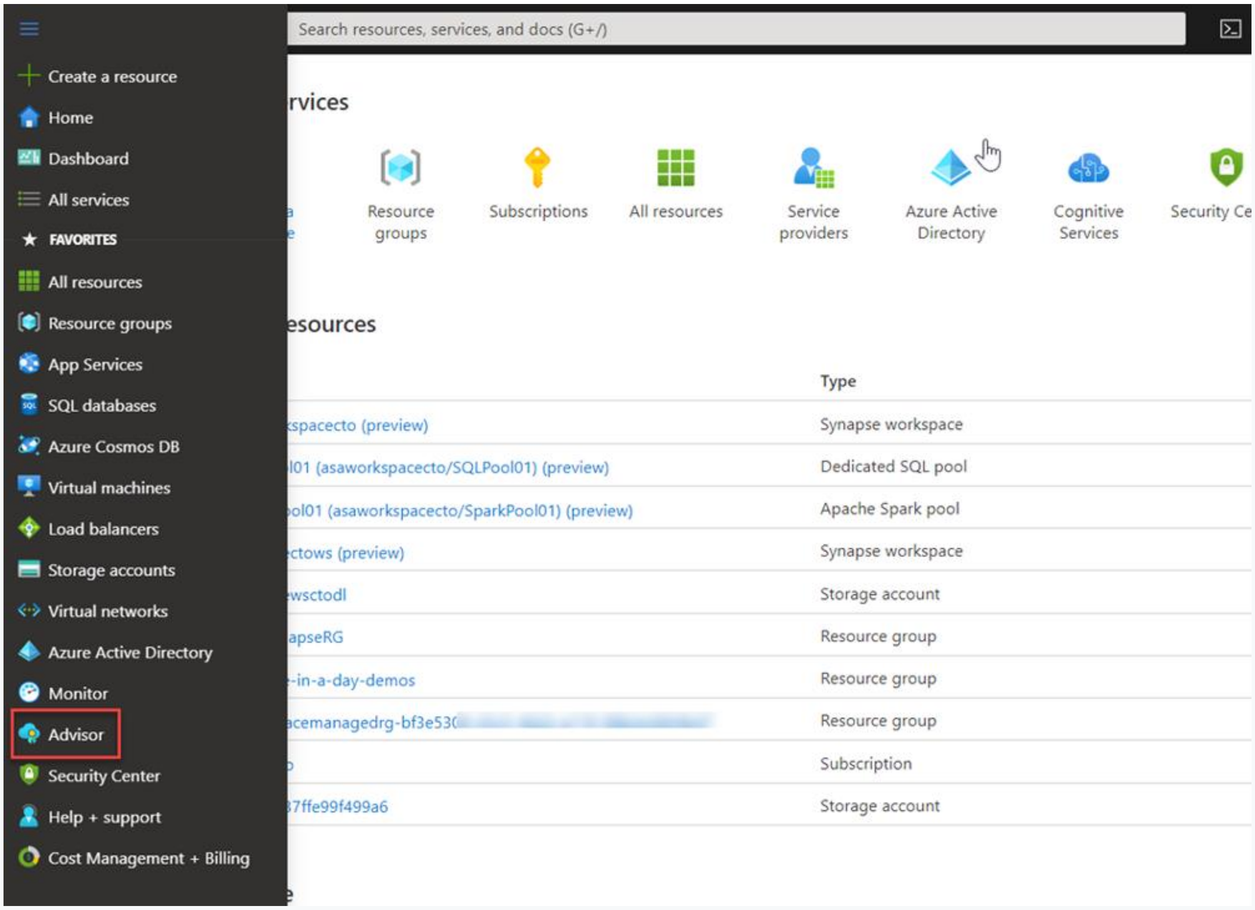
- □

  Reliability (formerly called High availability)

  (Correct)

Explanation
Azure Advisor provides you with personalized messages that provide information on best practices to optimize the setup of your Azure services. It analyzes your resource configuration and usage telemetry and then recommends solutions that can help you improve the cost effectiveness, performance, Reliability (formerly called High availability), and security of your Azure resources.

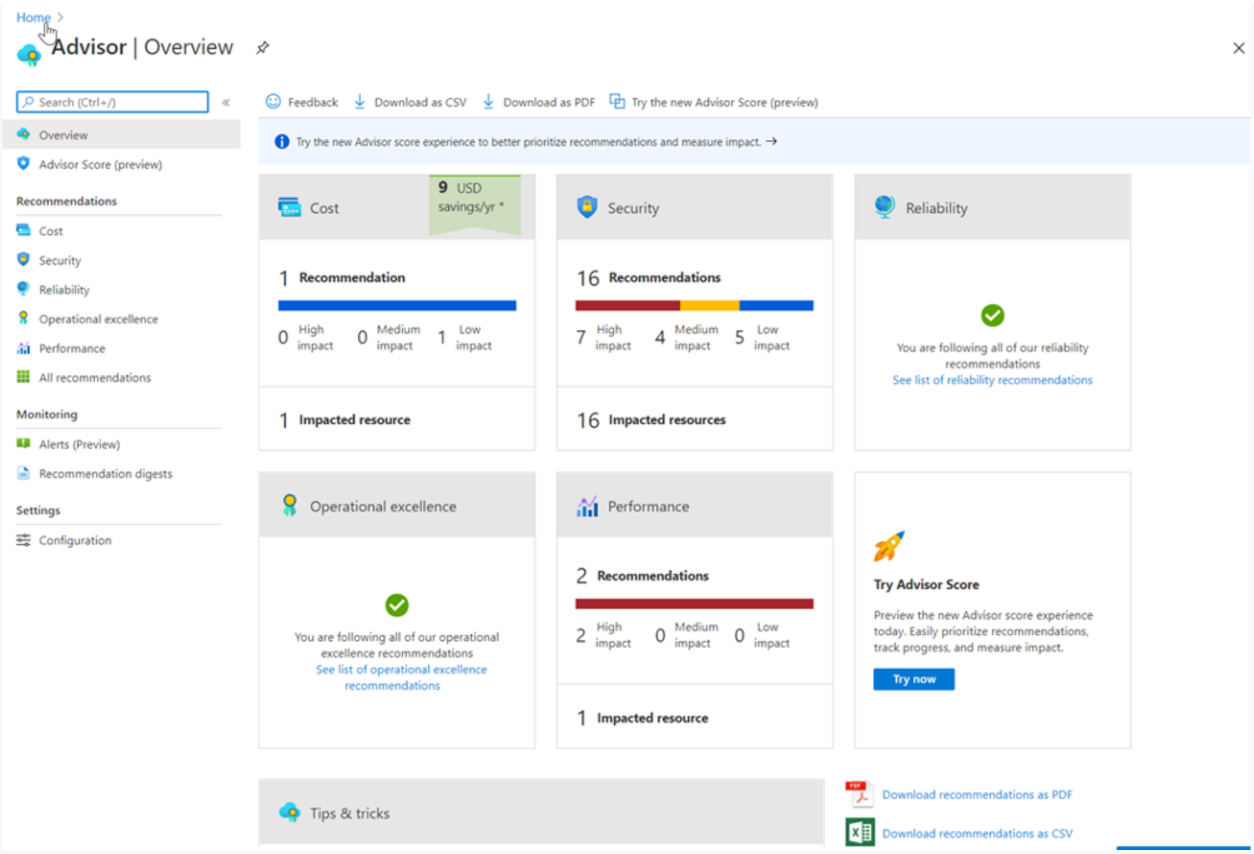
The Advisor may appear when you log into the Azure Portal, but you can also access the Advisor by selecting Advisor in the navigation menu.



	Type
kspacecto (preview)	Synapse workspace
l01 (asaworkspacecto/SQLPool01) (preview)	Dedicated SQL pool
ool01 (asaworkspacecto/SparkPool01) (preview)	Apache Spark pool
ctows (preview)	Synapse workspace
wsctodl	Storage account
apseRG	Resource group
-in-a-day-demos	Resource group
acemanagedrg-bf3e53[redacted]	Resource group
o	Subscription
37ffe99f499a6	Storage account

On accessing Advisor, a dashboard is presented that provides recommendations in the following areas:

- Cost
- Security
- Reliability
- Operational excellence
- Performance



Try the new Advisor score experience to better prioritize recommendations and measure impact. →

Cost: 1 Recommendation — 0 High impact, 0 Medium impact, 1 Low impact — 1 Impacted resource

Security: 16 Recommendations — 7 High impact, 4 Medium impact, 5 Low impact — 16 Impacted resources

Reliability: You are following all of our reliability recommendations. See list of reliability recommendations

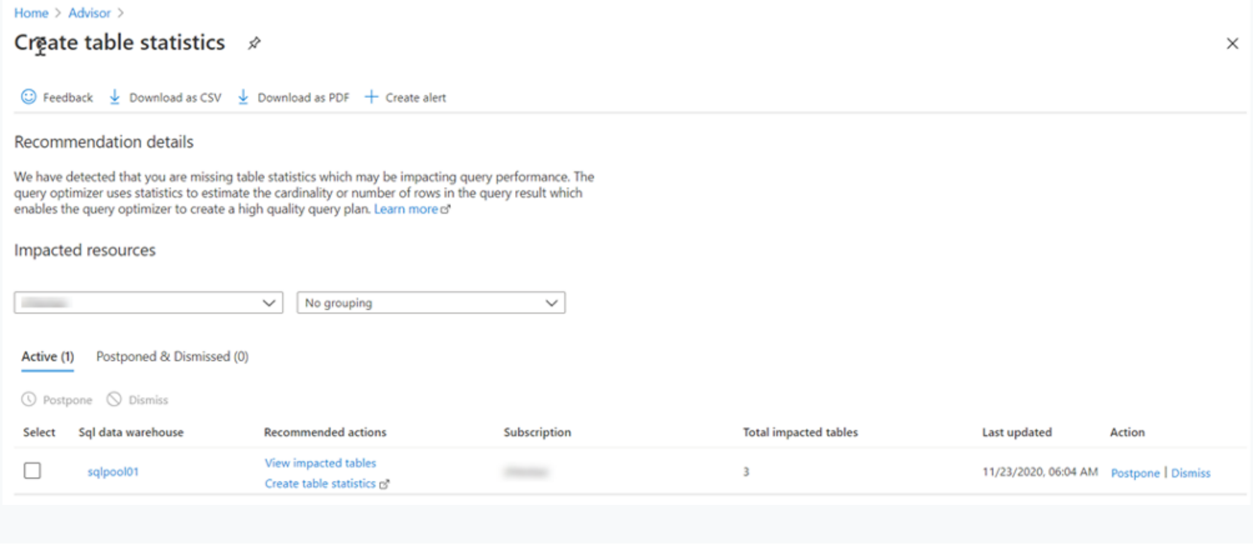
Operational excellence: You are following all of our operational excellence recommendations. See list of operational excellence recommendations

Performance: 2 Recommendations — 2 High impact, 0 Medium impact, 0 Low impact — 1 Impacted resource

Try Advisor Score

Preview the new Advisor score experience today. Easily prioritize recommendations, track progress, and measure impact.

You can click on any of the dashboard items for more information that can help you resolve the issue.



Recommendation details

We have detected that you are missing table statistics which may be impacting query performance. The query optimizer uses statistics to estimate the cardinality or number of rows in the query result which enables the query optimizer to create a high quality query plan. Learn more

Impacted resources

Select	Sql data warehouse	Recommended actions	Subscription	Total impacted tables	Last updated	Action
□	sqlpool01	View impacted tables Create table statistics	[redacted]	3	11/23/2020, 06:04 AM	Postpone | Dismiss

Once on this actual screen (not the image presented above), you can click on the **view impacted tables** to see which tables are being impacted specifically, and there are also links to the help in the Azure documentation that you can use to get more understanding of the issue.

https://docs.microsoft.com/en-us/azure/advisor/advisor-overview

Question 15: **Incorrect**

Identify the missing word(s) in the following sentence within the context of Microsoft Azure.

[?] logs every operation Azure Storage account activity in real time, and you can search the logs for specific requests. Filter based on the authentication mechanism, the success of the operation, or the resource that was accessed.

- ●

  Azure Advanced Threat Protection

  (Incorrect)

- ○

  Advanced Data Security

- ○

  Application Security Groups

- ○

  Network Security Groups

- ○

  Storage Analytics

  (Correct)

Explanation

Auditing access

Auditing is another part of controlling access. You can audit Azure Storage access by using the built-in Storage Analytics service.

Storage Analytics logs every operation in real time, and you can search the Storage Analytics logs for specific requests. Filter based on the authentication mechanism, the success of the operation, or the resource that was accessed.

https://docs.microsoft.com/en-us/azure/storage/common/storage-analytics

Question 16: **Incorrect**
Within the context of an Azure Databricks workspace, which command orders by a column in descending order?

- ○

  `df.orderBy("requests").show.desc()`

- ●

  `df.orderBy("requests").desc()`

  (Incorrect)

- ○

  `df.orderBy(col("requests").desc())`

  (Correct)

- ○

  `df.orderBy("requests desc")`

Explanation
Use the `.desc()` method on the Column Class to reverse the order.

https://sparkbyexamples.com/pyspark/pyspark-orderby-and-sort-explained/

Question 17: **Incorrect**
Scenario: You are working as a consultant at Avengers Security. At the moment, you are consulting with Tony, the lead of the IT team and the topic of discussion is about altering a table in an Azure Synapse Analytics dedicated SQL pool based on some specific requirements.

Happy Hogan, the lead developer, has created a table by using the following Transact-SQL statement.

```
CREATE TABLE [dbo].[DimEmployee](
    [EmployeeKey] [int] IDENTITY (1,1) NOT NULL,
    [EmployeeID] [int] NOT NULL,
    [FirstName] [varchar] (100) NOT NULL,
    [LastName] [varchar] (100) NOT NULL,
    [JobTitle] [varchar] (100) NOT NULL,
    [LastHireDate] [date] NULL,
    [StreetAddress] [varchar] (500) NOT NULL,
    [City] [varchar] (200) NOT NULL,
    [ProvinceState] [varchar] (50) NOT NULL,
    [PostalCode] [varchar] (10) NOT NULL,
    )
```

Required:

The table must be altered to meet the following items:

1. It must ensure that users can identify the current manager of any employee.

2. It must support creating an employee reporting hierarchy for the entire company.

3. It must provide a simple lookup of the managers' attributes (name and job title).

Which column should be added to the table?

- ●

  `[ManagerName] [varchar](200) NULL`

  (Incorrect)

- ○

  `[DimEmployee] [int] NULL`

- ○

  `[ManagerEmployeeKey] [int] NULL`

  (Correct)

- ○

  `[ManagerEmployeeID] [int] NULL`

Explanation

`[ManagerEmployeeKey] [int] NULL` is the correct line to add to the table. In dimensions we use surrogates. If `[ManagerEmployeeID] [int] NULL` is used to create a hierarchy, at the time of the insert we can't guarantee that the manager is already inserted and thus we can't resolve the `EmployeeKey` of the manager, because it is an identity.

Hierarchies, in tabular models, are metadata that define relationships between two or more columns in a table. Hierarchies can appear separate from other columns in a reporting client field list, making them easier for client users to navigate and include in a report.

Benefits

Tables can include dozens or even hundreds of columns with unusual column names in no apparent order. This can lead to an unordered appearance in reporting client field lists, making it difficult for users to find and include data in a report. Hierarchies can provide a simple, intuitive view of an otherwise complex data structure.

For example, in a Date table, you can create a Calendar hierarchy. Calendar Year is used as the top-most parent level, with Month, Week, and Day included as child levels (Calendar Year->Month->Week->Day). This hierarchy shows a logical relationship from Calendar Year to Day. A client user can then select Calendar Year from a Field List to include all levels in a PivotTable, or expand the hierarchy, and select only particular levels to be included in the PivotTable.

Because each level in a hierarchy is a representation of a column in a table, the level can be renamed. While not exclusive to hierarchies (any column can be renamed in a tabular model), renaming hierarchy levels can make it easier for users to find and include levels in a report. Renaming a level does not rename the column it references; it simply makes the level more identifiable. In our Calendar Year hierarchy example, in the Date table in Data View, the columns: CalendarYear, CalendarMonth, CalendarWeek, and CalendarDay were renamed to Calendar Year, Month, Week, and Day to make them more easily identifiable. Renaming levels has the additional benefit of providing consistency in reports, since users will less likely need to change column names to make them more readable in PivotTables, charts, etc.

Hierarchies can be included in perspectives. Perspectives define viewable subsets of a model that provide focused, business-specific, or application-specific viewpoints of the model. A perspective, for example, could provide users a viewable list (hierarchy) of only those data items necessary for their specific reporting requirements. For more information, see Perspectives.

Hierarchies are not meant to be used as a security mechanism, but as a tool for providing a better user experience. All security for a particular hierarchy is inherited from the underlying model. Hierarchies cannot provide access to model objects to which a user does not already have access. Security for the model database must be resolved before access to objects in the model can be provided through a hierarchy.

Security roles can be used to secure model metadata and data. For more information, see Roles.

Defining hierarchies

You create and manage hierarchies by using the model designer in Diagram View. Creating and managing hierarchies is not supported in the model designer in Data View. To view the model designer in Diagram View, click the **Model** menu, then point to **Model View**, and then click **Diagram View**.

To create a hierarchy, right-click a column you want to specify as the parent level, and then click **Create Hierarchy**. You can multi-select any number of columns (within a single table) to include, or you can later add columns as child levels by clicking and dragging columns to the parent level. When multiple columns are selected, columns are automatically placed in an order based on cardinality. You can change the order by clicking and dragging a column (level) to a different order or by using Up and Down navigation controls on the context menu. When adding a column as a child level, you can use auto-detect by dragging and dropping the column onto the parent level.

A column can appear in more than one hierarchy. Hierarchies cannot include non-column objects such as measures or KPIs. A hierarchy can be based on columns from within a single table only. If you multi-select a measure along with one or more columns, or if you select columns from multiple tables, the **Create Hierarchy** command is disabled in the context menu. To add a column from a different table, use the RELATED DAX function to add a calculated column that references the column from the related table. The function uses the following syntax: `=RELATED(TableName[ColumnName])`. For more information, see RELATED Function.

By default, new hierarchies are named hierarchy1, hierarchy 2, etc. You should change hierarchy names to reflect the nature of the parent-child relationship, making them more identifiable to users.

https://docs.microsoft.com/en-us/analysis-services/tabular-models/hierarchies-ssas-tabular?view=asallproducts-allversions

Question 18: **Correct**
Scenario: You need an NoSQL database of a supported API model, at planet scale, and with low latency performance.

Which of the following should you choose?

- ○

  Azure Database for MySQL

- ●

  Azure Cosmos DB

  (Correct)

- ○

  Azure DB for PostgreSQL Single Server

- ○

  Azure DB for MySQL Single Server

- ○

  Azure DB Server

- ○

  Azure Database for MariaDB

- ○

  Azure Database for PostgreSQL

Explanation

When to use Azure Cosmos DB

Deploy Azure Cosmos DB when you need a NoSQL database of the supported API model, at planet scale, and with low latency performance. Currently, Azure Cosmos DB supports five-nines uptime (99.999 percent). It can support response times below 10 ms when it's provisioned correctly.

https://azure.microsoft.com/en-us/services/cosmos-db/

Question 19: **Incorrect**

Azure Databricks is an amalgamation of multiple technologies that enable you to work with data at scale.

Which of the following is best described by *"A distributed data processing solution that makes use of clusters to scale processing across multiple compute nodes. Each cluster has a driver node to coordinate processing jobs, and one or more worker nodes on which the processing occurs. This distributed model enables each node to operate on a subset of the job in parallel; reducing the overall time for the job to complete"*?

- ●

  Hive metastore clusters

  (Incorrect)

- ○

  Apache Spark clusters

  (Correct)

- ○

  Notebook clusters

- ○

  Databricks File System clusters

- ○

  Delta Lake clusters

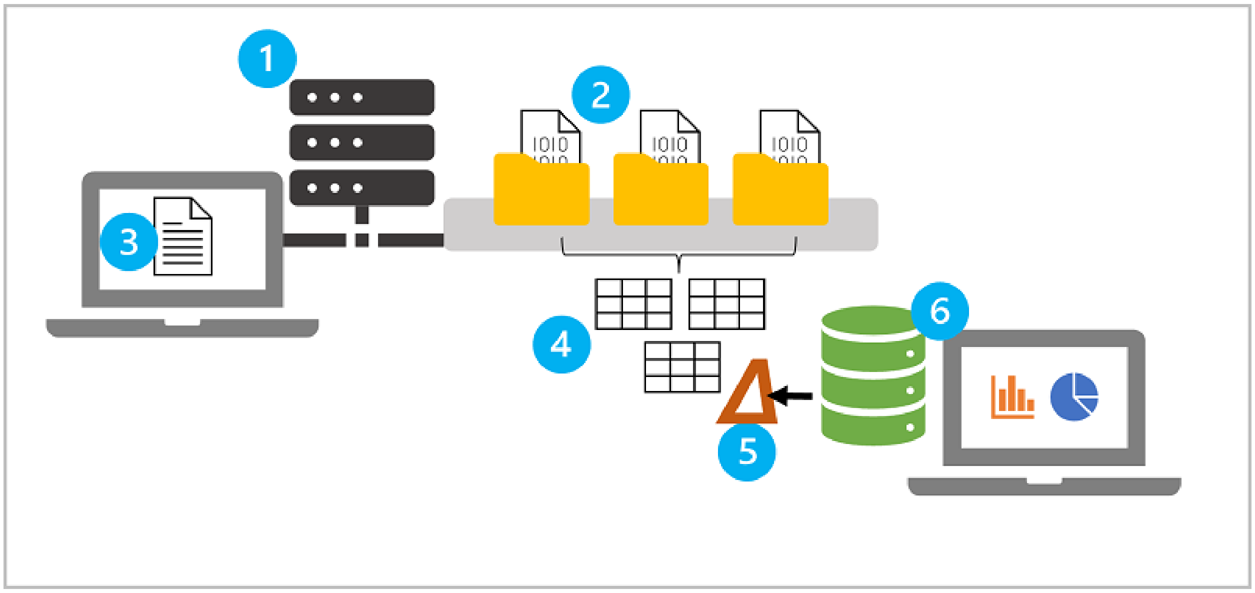
- ○

  SQL Warehouses

Explanation

Understand key concepts

Azure Databricks is an amalgamation of multiple technologies that enable you to work with data at scale. Before using Azure Databricks, there are some key concepts that you should understand.



1. **Apache Spark clusters** - Spark is a distributed data processing solution that makes use of *clusters* to scale processing across multiple compute *nodes*. Each Spark cluster has a *driver* node to coordinate processing jobs, and one or more *worker* nodes on which the processing occurs. This distributed model enables each node to operate on a subset of the job in parallel; reducing the overall time for the job to complete.

https://learn.microsoft.com/en-us/azure/databricks/clusters/

2. Databricks File System (DBFS) - While each cluster node has its own local file system (on which operating system and other node-specific files are stored), the nodes in a cluster have access to a shared, distributed file system in which they can access and operate on data files. The *Databricks File System* (DBFS) enables you to mount cloud storage and use it to work with and persist file-based data.

https://learn.microsoft.com/en-us/azure/databricks/data/databricks-file-system

3. Notebooks - One of the most common ways for data analysts, data scientists, data engineers, and developers to work with Spark is to write code in *notebooks*. Notebooks provide an interactive environment in which you can combine text and graphics in *Markdown* format with cells containing code that you run interactively in the notebook session.

https://learn.microsoft.com/en-us/azure/databricks/notebooks/

4. Hive metastore - *Hive* is an open-source technology used to define a relational abstraction layer of tables over file-based data. The tables can then be queried using SQL syntax. The table definitions and details of the file system locations on which they're based is stored in the metastore for a Spark cluster. A *Hive metastore* is created for each cluster when it's created, but you can configure a cluster to use an existing external metastore if necessary.

https://learn.microsoft.com/en-us/azure/databricks/data/metastores/

5. Delta Lake - *Delta Lake* builds on the relational table schema abstraction over files in the data lake to add support for SQL semantics commonly found in relational database systems. Capabilities provided by Delta Lake include transaction logging, data type constraints, and the ability to incorporate streaming data into a relational table.

https://learn.microsoft.com/en-us/azure/databricks/delta/

6. SQL Warehouses - *SQL Warehouses* are relational compute resources with endpoints that enable client applications to connect to an Azure Databricks workspace and use SQL to work with data in tables. The results of SQL queries can be used to create data visualizations and dashboards to support business analytics and decision making. SQL Warehouses are only available in *premium* tier Azure Databricks workspaces.

https://learn.microsoft.com/en-us/azure/databricks/sql/admin/sql-endpoints

Question 20: **Incorrect**
In Azure Synapse Studio, the Develop hub is where you access which of the following? (Select four)

- □

  Power BI

  (Correct)

- ☑

  Notebooks

  (Correct)

- □

  Master Pipeline

- □

  SQL scripts

  (Correct)

- □

  Activities

- □

  External data sources

- □

  Data flows

  (Correct)

- □

  Pipeline canvas

- □

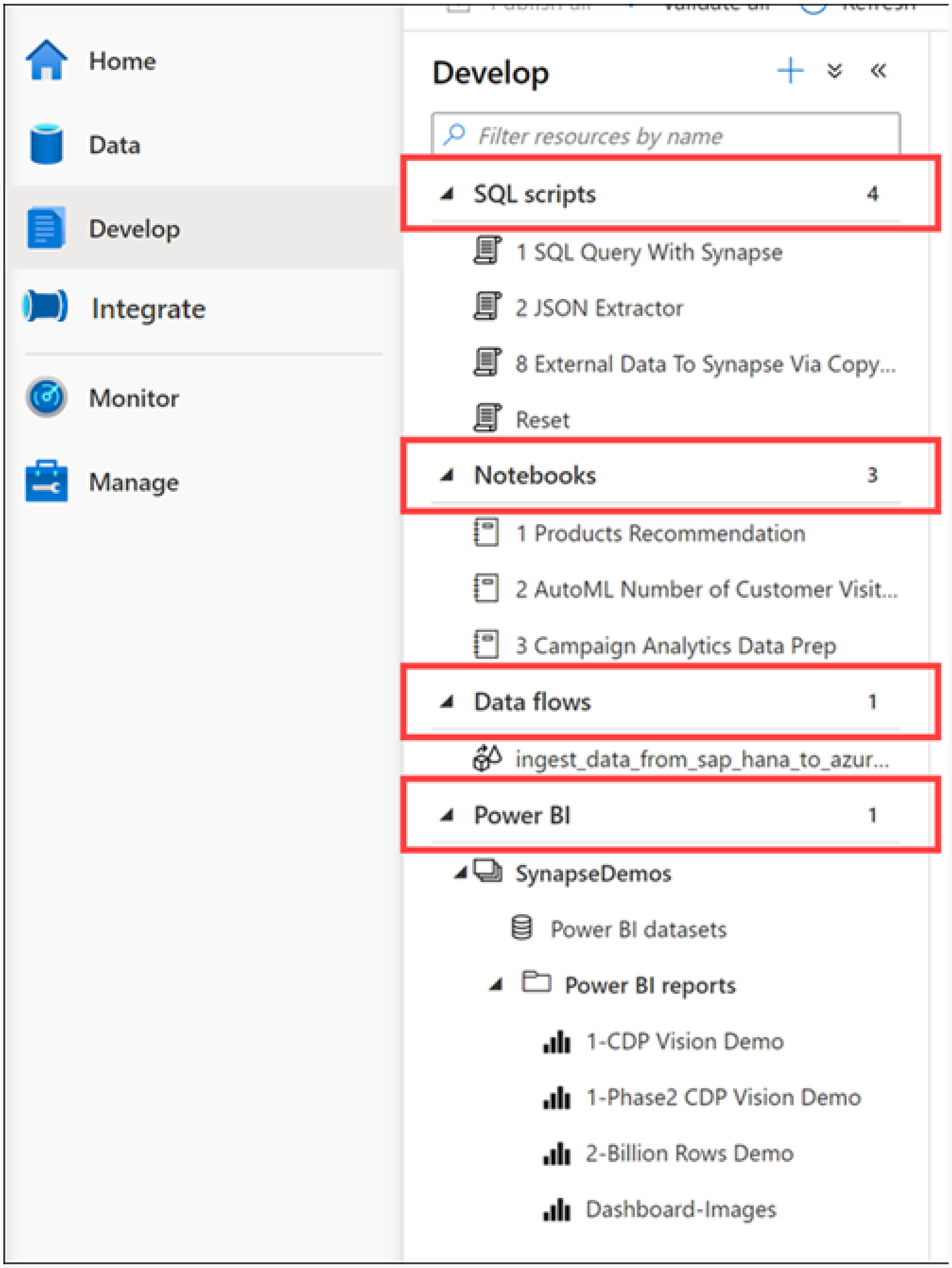
  SQL serverless databases

- □

  Provisioned SQL pool databases

Explanation

In Azure Synapse Studio, the Develop hub is where you manage SQL scripts, Synapse notebooks, data flows, and Power BI reports.



The Develop hub in our sample environment contains examples of the following artifacts:

- **SQL scripts** contains T-SQL scripts that you publish to your workspace. Within the scripts, you can execute commands against any of the provisioned SQL pools or on-demand SQL serverless pools to which you have access.

• **Notebooks** contains Synapse Spark notebooks used for data engineering and data science tasks. When you execute a notebook, you select a Spark pool as its compute target.

• **Data flows** are powerful data transformation workflows that use the power of Apache Spark but are authored using a code-free GUI.

• **Power BI** reports can be embedded here, giving you access to the advanced visualizations they provide without ever leaving the Synapse workspace.

https://www.techtalkcorner.com/azure-synapse-analytics-develop-hub/

Question 21: **Incorrect**

Query languages used in Synapse SQL can have different supported features depending on consumption model.

Which of the following are compatible with the Dedicated consumption model?

- ☑

  `UPDATE` statement

  (Correct)

- □

  Control of flow

  (Correct)

- □

  `MERGE` statement

  (Correct)

- □

  `SELECT` statement

  (Correct)

- □

Data export

(Correct)

- □

  DDL statements (`CREATE`, `ALTER`, `DROP`)

  (Correct)

- □

  `DELETE` statement

  (Correct)

- □

  `INSERT` statement

  (Correct)

- □

  Cross-database queries

- □

  Built-in functions (analysis)

  (Correct)

Explanation
Azure Synapse Analytics supports querying both relational (dedicated and serverless SQL endpoints) and non-relational data (Azure Data Lake Storage Gen 2, Cosmos DB and Azure Blob Storage) at petabyte-scale using Transact SQL, supporting ANSI-compliant SQL language.

The Azure Synapse SQL query language supports different features based on the resource model being used. The table below outlines which Transact-SQL statements work against each resource model.

	Dedicated	Serverless
SELECT statement	Yes. Transact-SQL query clauses FOR XML/FOR JSON, and MATCH are not supported.	Yes. Transact-SQL query clauses FOR XML, MATCH, PREDICT, and query hints are not supported.
INSERT statement	Yes	No
UPDATE statement	Yes	No
DELETE statement	Yes	No
MERGE statement	Yes (preview)	No
Transactions	Yes	Yes, applicable on meta-data objects.
Labels	Yes	No
Data load	Yes. Preferred utility is COPY statement, but the system supports both BULK load (BCP) and CETAS for data loading.	No
Data export	Yes. Using CETAS.	Yes. Using CETAS.
Types	Yes, all Transact-SQL types except cursor, hierarchyid, ntext, text, and image, rowversion, Spatial Types, sql_variant, and xml	Yes, all Transact-SQL types except cursor, hierarchyid, ntext, text, and image, rowversion, Spatial Types, sql_variant, xml, and Table type
Cross-database queries	No	Yes, including USE statement.
Built-in functions (analysis)	Yes, all Transact-SQL Analytic, Conversion, Date and Time, Logical, Mathematical functions, except CHOOSE, IIF, and PARSE	Yes, all Transact-SQL Analytic, Conversion, Date and Time, Logical, Mathematical functions.
Built-in functions (text)	Yes. All Transact-SQL String, JSON, and Collation functions, except STRING_ESCAPE and TRANSLATE	Yes. All Transact-SQL String, JSON, and Collation functions.
Built-in table-value functions	Yes, Transact-SQL Rowset functions, except OPENXML, OPENDATASOURCE, OPENQUERY, and OPENROWSET	Yes, Transact-SQL Rowset functions, except OPENXML, OPENDATASOURCE, and OPENQUERY
Aggregates	Transact-SQL built-in aggregates except, except CHECKSUM_AGG and GROUPING_ID	Transact-SQL built-in aggregates.
Operators	Yes, all Transact-SQL operators except !> and !<	Yes, all Transact-SQL operators
Control of flow	Yes. All Transact-SQL Control-of-flow statement except CONTINUE, GOTO, RETURN, USE, and WAITFOR	Yes. All Transact-SQL Control-of-flow statement SELECT query in WHILE (...) condition
DDL statements (CREATE, ALTER, DROP)	Yes. All Transact-SQL DDL statement applicable to the supported object types	Yes. All Transact-SQL DDL statement applicable to the supported object types

https://docs.microsoft.com/en-us/azure/synapse-analytics/sql/overview-features

Question 22: **Correct**

Scenario: O'Shaughnessy's is a fast food restaurant. The chain has stores nationwide and is rivalled by Big Belly Burgers. You have been hired by the company to advise on working with Microsoft Azure Data Lake Storage.

At the moment, the team is planning the deployment of Azure Data Lake Storage Gen2.

There are two reports which will access the data lake:

• **Report1:** Reads three columns from a file that contains 50 columns.

• **Report2:** Queries a single record based on a timestamp.

As the Azure expert, the team is looking for you to recommend in which format to store the data in the data lake to support the reports. The solution must minimize read times.

The options available are:

a. AVRO

b. CSV

c. Parquet

d. TSV

Which should you recommend to O'Shaughnessy's for each report?

- ●

  Report1: Parquet, Report2: AVRO

  (Correct)

- ○

  Report1: CSV, Report2: Parquet

- ○

  Report1: CSV, Report2: TSV

- ○

Report1: Parquet, Report2: TSV

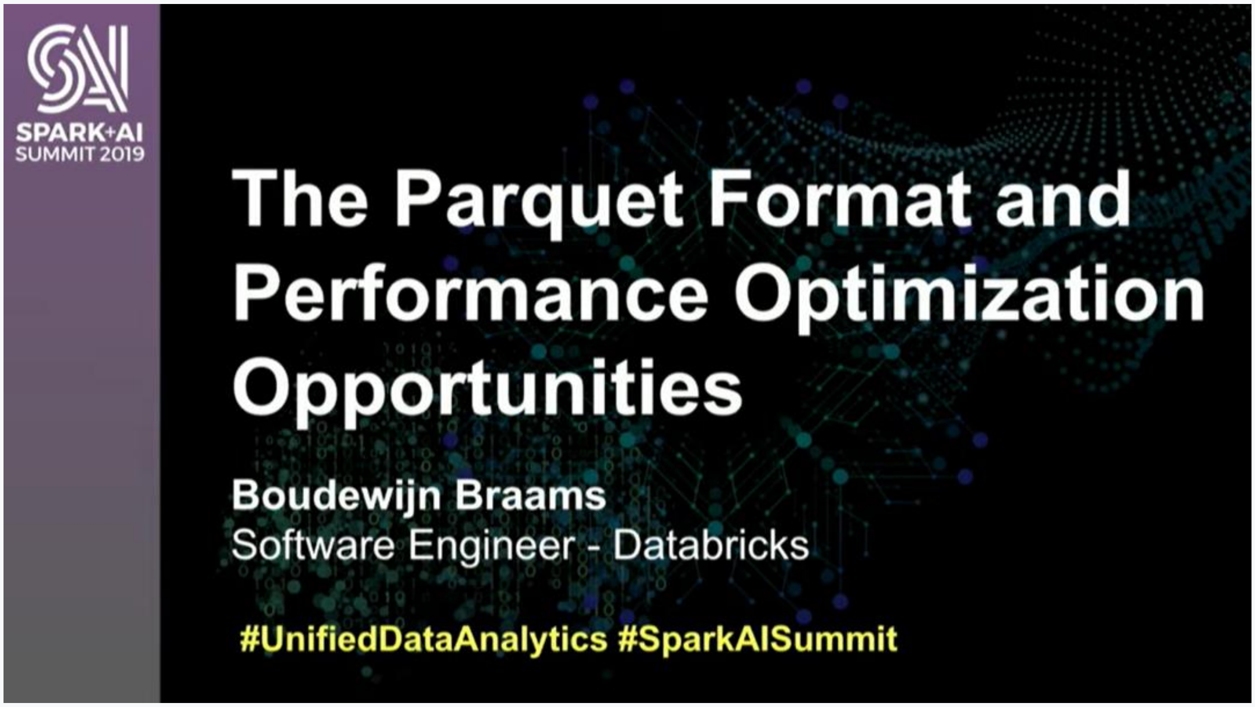
Explanation

Report1: Parquet - hybrid model suited for both OLTP and OLAP.

Parquet format is supported for the following connectors: Amazon S3, Amazon S3 Compatible Storage, Azure Blob, Azure Data Lake Storage Gen1, Azure Data Lake Storage Gen2, Azure File Storage, File System, FTP, Google Cloud Storage, HDFS, HTTP, Oracle Cloud Storage and SFTP.

For a list of supported features for all available connectors, visit the Connectors Overview article.

https://docs.microsoft.com/en-us/azure/data-factory/format-parquet


SPARK+AI
SUMMIT 2019

The Parquet Format and Performance Optimization Opportunities

Boudewijn Braams

Software Engineer - Databricks

#UnifiedDataAnalytics #SparkAISummit

https://www.youtube.com/watch?v=1j8SdS7s_NY

Report2: AVRO - Row based format, and has logical type timestamp

The Azure Data Lake Storage Gen2 destination writes data to Azure Data Lake Storage Gen2 based on the data format that you select. You can use the following data formats:

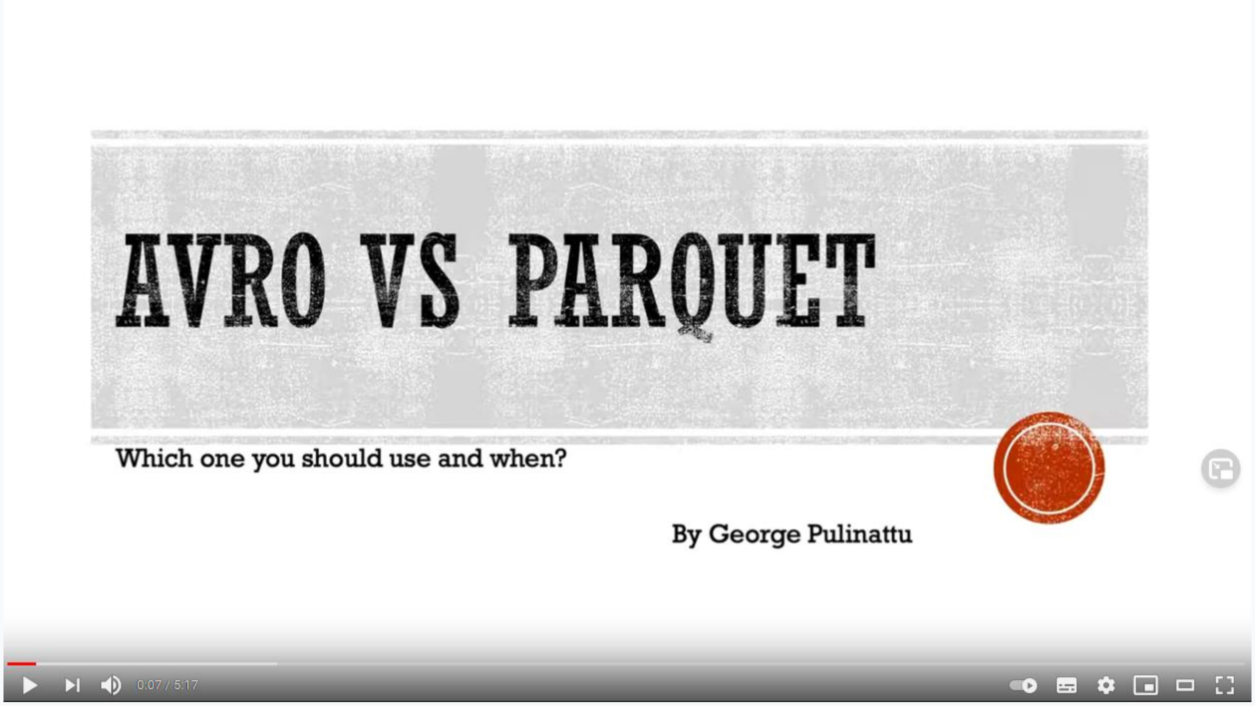
Avro The destination writes records based on the Avro schema. You can use one of the following methods to specify the location of the Avro schema definition: In Pipeline

Configuration - Use the schema that you provide in the stage configuration. In Record Header - Use the schema included in the avroSchema record header attribute. Confluent Schema Registry - Retrieve the schema from Confluent Schema Registry. Confluent Schema Registry is a distributed storage layer for Avro schemas. You can configure the destination to look up the schema in Confluent Schema Registry by the schema ID or subject.

If using the Avro schema in the stage or in the record header attribute, you can optionally configure the destination to register the Avro schema with Confluent Schema Registry.

The destination includes the schema definition in each file. You can compress data with an Avro-supported compression codec. When using Avro compression, avoid using other compression properties in the destination.

https://streamsets.com/documentation/datacollector/latest/help/datacollector/UserGuide/Destinations/ADLS-G2-D.html



AVRO VS PARQUET

Which one you should use and when?

By George Pulinattu

https://youtu.be/UrWthx8T3UY

Question 23: **Incorrect**

Azure Databricks includes an integrated notebook interface for working with Spark. Notebooks provide an intuitive way to combine code with Markdown notes, commonly used by data scientists and data analysts.

Notebooks consist of one or more cells, each containing either code or markdown. Code cells in notebooks have some features that can help you be more productive.

Which of the following are valid features of notebooks when using Spark? (Select four)

- ☑

  Syntax highlighting and error support.

  (Correct)

- □

  The ability to export results.

  (Correct)

- □

  Code auto-completion.

  (Correct)

- □

  Develop code using Python, SQL, Scala, C#, and R.

- □

  Interactive data visualizations.

  (Correct)

- □

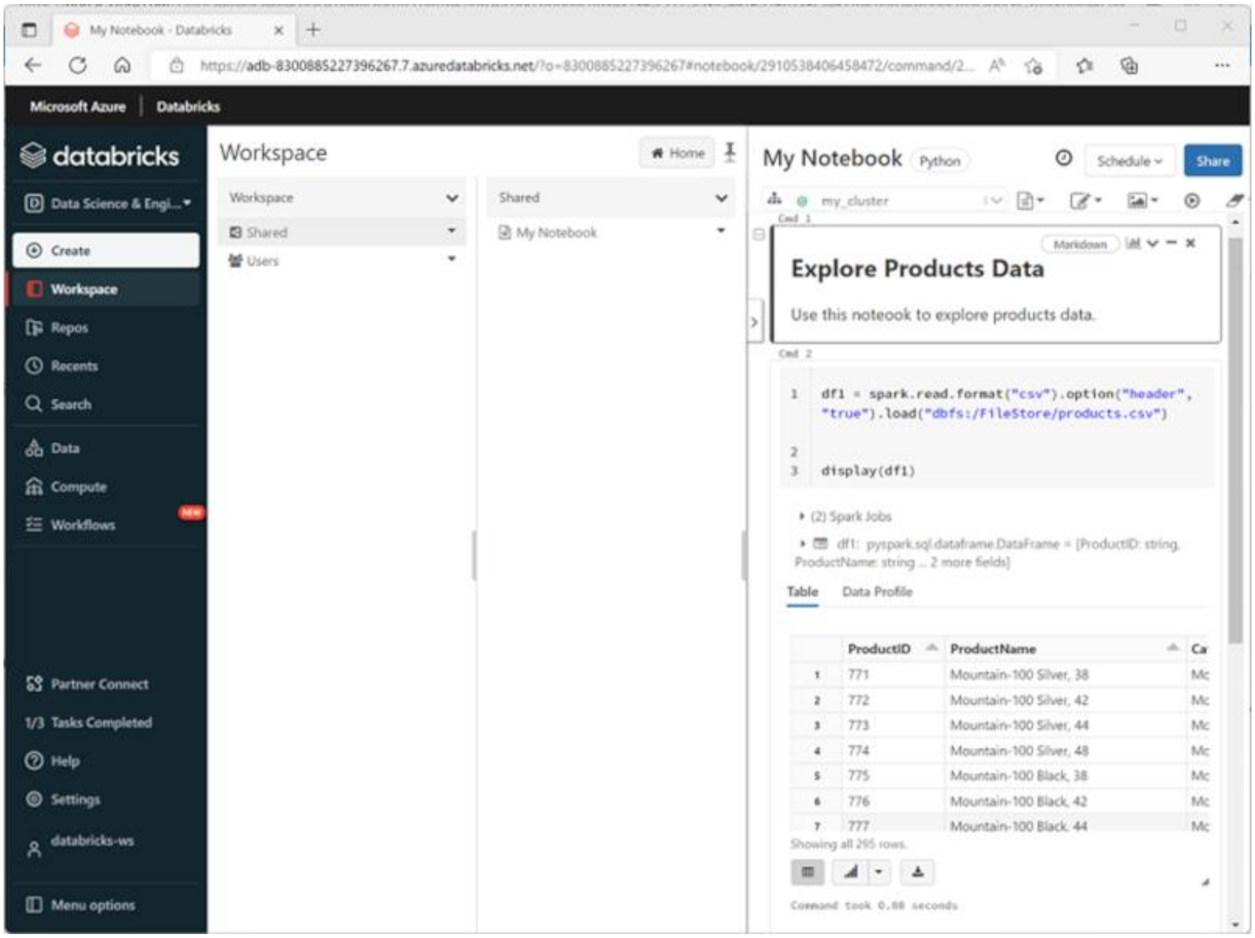
  Export results and notebooks in .html, .xml, or .ipynb format.

Explanation

Use Spark in notebooks

Running Spark code in notebooks

Azure Databricks includes an integrated notebook interface for working with Spark. Notebooks provide an intuitive way to combine code with Markdown notes, commonly used by data scientists and data analysts. The look and feel of the integrated notebook experience within Azure Databricks is similar to that of Jupyter notebooks - a popular open-source notebook platform.



Explore Products Data

Use this noteook to explore products data.

```
df1 = spark.read.format("csv").option("header", "true").load("dbfs:/FileStore/products.csv")

display(df1)
```

(2) Spark Jobs

df1: pyspark.sql.dataframe.DataFrame = [ProductID: string, ProductName: string ... 2 more fields]

	ProductID	ProductName	Ca
1	771	Mountain-100 Silver, 38	Mc
2	772	Mountain-100 Silver, 42	Mc
3	773	Mountain-100 Silver, 44	Mc
4	774	Mountain-100 Silver, 48	Mc
5	775	Mountain-100 Black, 38	Mc
6	776	Mountain-100 Black, 42	Mc
7	777	Mountain-100 Black, 44	Mc

Showing all 295 rows.

Command took 0.88 seconds

Notebooks consist of one or more cells, each containing either code or markdown. Code cells in notebooks have some features that can help you be more productive, including:

- Syntax highlighting and error support.

- Code auto-completion.

- Interactive data visualizations.

- The ability to export results.

With Azure Databricks notebooks, you can:

- Develop code using Python, SQL, Scala, and R.
- Customize your environment with the libraries of your choice.
- Create regularly scheduled jobs to automatically run tasks, including multi-notebook workflows.
- Export results and notebooks in `.html` or `.ipynb` format.
- Use a Git-based repository to store your notebooks with associated files and dependencies.
- Build and share dashboards.
- Open or run a Delta Live Tables pipeline.
- (Experimental) Use advanced editing capabilities.

https://learn.microsoft.com/en-us/azure/databricks/notebooks/

Question 24: **Incorrect**
Identify the missing word(s) in the following sentence within the context of Microsoft Azure.

From a high level, the Azure Databricks service launches and manages Apache Spark clusters within your Azure subscription. Apache Spark clusters are groups of computers that are treated as a single computer and handle the execution of commands issued from notebooks.

Microsoft Azure manages the cluster, and [?]

- ●

  specifies the types and sizes of the virtual machines.

  (Incorrect)

- ○

  provides the fastest virtualized network infrastructure in the cloud.

- ○

  auto-scales it as needed based on your usage and the setting used when configuring the cluster.

  (Correct)

- ○

pulls data from a specified data source.

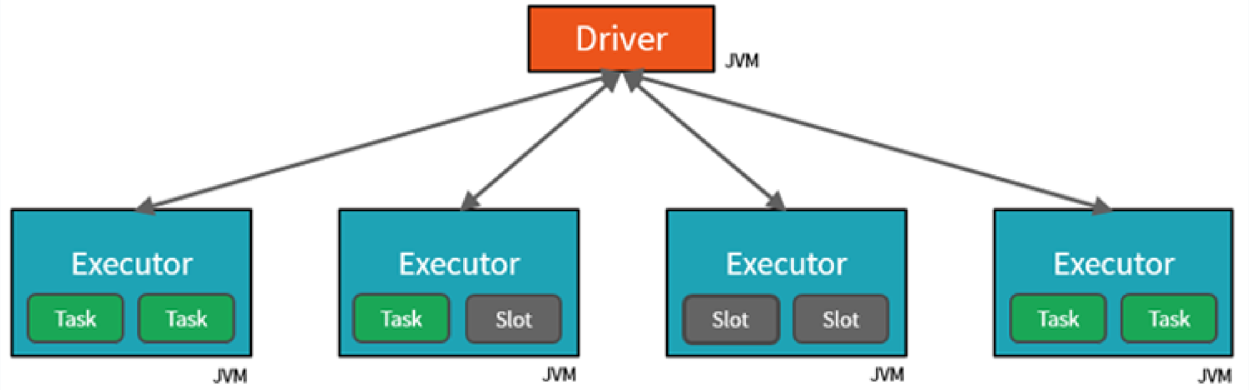
Explanation

To gain a better understanding of how to develop with Azure Databricks, it is important to understand the underlying architecture. We will look at two aspects of the Databricks architecture: the Azure Databricks service and Apache Spark clusters.

High-level overview

From a high level, the Azure Databricks service launches and manages Apache Spark clusters within your Azure subscription. Apache Spark clusters are groups of computers that are treated as a single computer and handle the execution of commands issued from notebooks. Using a master-worker type architecture, clusters allow processing of data to be parallelized across many computers to improve scale and performance. They consist of a Spark Driver (master) and worker nodes. The driver node sends work to the worker nodes and instructs them to pull data from a specified data source.

In Databricks, the notebook interface is the driver program. This driver program contains the main loop for the program and creates distributed datasets on the cluster, then applies operations (transformations & actions) to those datasets. Driver programs access Apache Spark through a SparkSession object regardless of deployment location.



Microsoft Azure manages the cluster, and auto-scales it as needed based on your usage and the setting used when configuring the cluster. Auto-termination can also be enabled, which allows Azure to terminate the cluster after a specified number of minutes of inactivity.

Under the covers

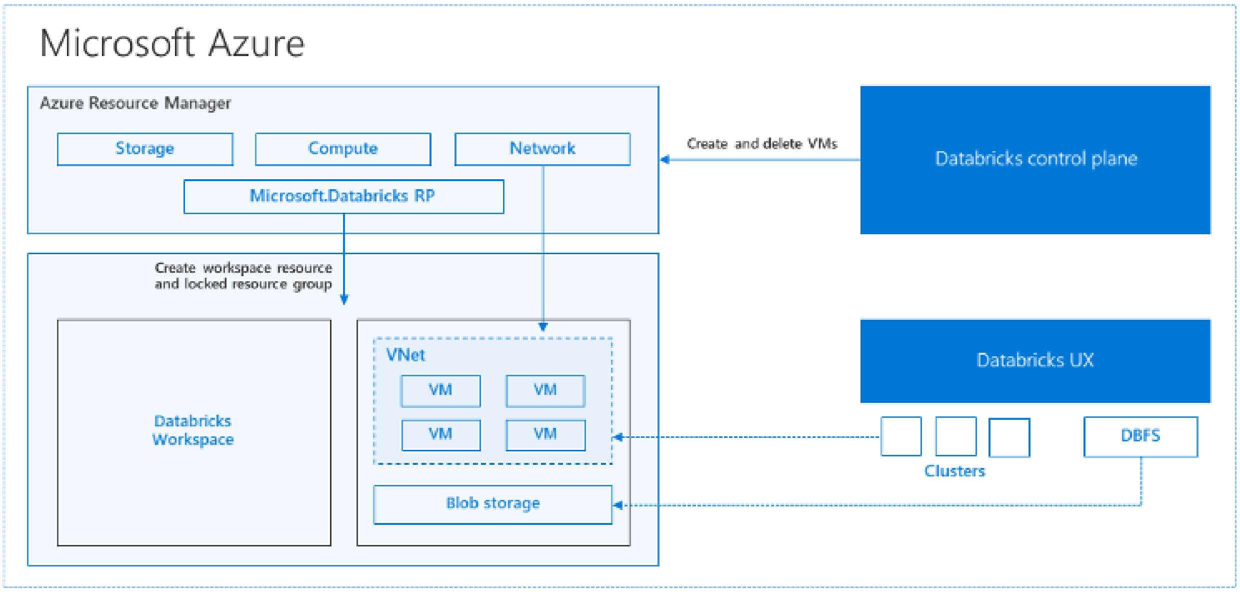
Now let's take a deeper look under the covers. When you create an Azure Databricks service, a "Databricks appliance" is deployed as an Azure resource in your subscription.

At the time of cluster creation, you specify the types and sizes of the virtual machines (VMs) to use for both the Driver and Worker nodes, but Azure Databricks manages all other aspects of the cluster.

You also have the option of using a Serverless Pool. A Serverless Pool is self-managed pool of cloud resources that is auto-configured for interactive Spark workloads. You provide the minimum and maximum number of workers and the worker type, and Azure Databricks provisions the compute and local storage based on your usage.

The "Databricks appliance" is deployed into Azure as a managed resource group within your subscription. This resource group contains the Driver and Worker VMs, along with other required resources, including a virtual network, a security group, and a storage account. All metadata for your cluster, such as scheduled jobs, is stored in an Azure Database with geo-replication for fault tolerance.

Internally, Azure Kubernetes Service (AKS) is used to run the Azure Databricks control-plane and data-planes via containers running on the latest generation of Azure hardware (Dv3 VMs), with NvMe SSDs capable of blazing 100us latency on IO. These make Databricks I/O performance even better. In addition, accelerated networking provides the fastest virtualized network infrastructure in the cloud. Azure Databricks utilizes these features to further improve Spark performance. Once the services within this managed resource group are ready, you will be able to manage the Databricks cluster through the Azure Databricks UI and through features such as auto-scaling and auto-termination.



https://databricks.com/blog/2017/11/15/a-technical-overview-of-azure-databricks.html

Question 25: **Incorrect**

You can create one or more clusters in your Azure Databricks workspace by using the Azure Databricks portal. When creating the cluster, you can specify configuration settings.

Which of the following are valid options for a *cluster mode*? (Select three)

- ☑

  Single Node

  (Correct)

- □

  Standard

  (Correct)

- □

  Default

- □

  High Concurrency

  (Correct)

- □

  Low Concurrency

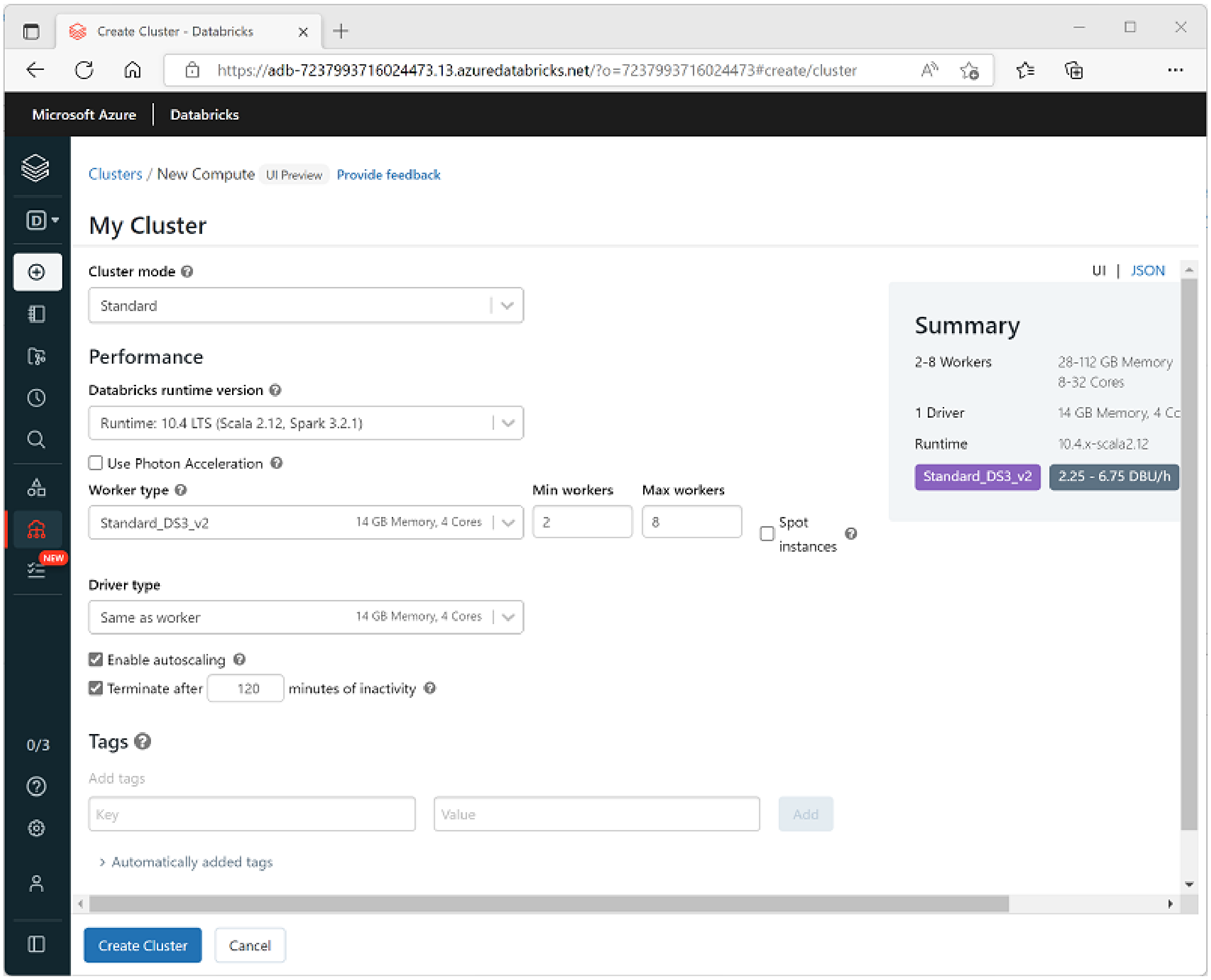
- □

  Multi-node

Explanation

Create a Spark cluster

You can create one or more clusters in your Azure Databricks workspace by using the Azure Databricks portal.


Create Cluster - Databricks
https://adb-7237993716024473.13.azuredatabricks.net/?o=7237993716024473#create/cluster
Microsoft Azure
Databricks
UI | JSON
NEW
0/3
Create Cluster
Cancel

Clusters / New Compute UI Preview Provide feedback

My Cluster

Cluster mode: Standard

Performance

Databricks runtime version: Runtime: 10.4 LTS (Scala 2.12, Spark 3.2.1)

☐ Use Photon Acceleration

Worker type: Standard_DS3_v2 14 GB Memory, 4 Cores

Min workers: 2

Max workers: 8

☐ Spot instances

Driver type: Same as worker 14 GB Memory, 4 Cores

☑ Enable autoscaling

☑ Terminate after 120 minutes of inactivity

Tags

Add tags

Key

Value

Add

› Automatically added tags

Summary

2-8 Workers	28-112 GB Memory 8-32 Cores
1 Driver	14 GB Memory, 4 Cc
Runtime	10.4.x-scala2.12
Standard_DS3_v2	2.25 - 6.75 DBU/h

When creating the cluster, you can specify configuration settings, including:

- A name for the cluster.

- A *cluster mode*, which can be:

  - *Standard*: Suitable for single-user workloads that require multiple worker nodes.

  - *High Concurrency*: Suitable for workloads where multiple users will be using the cluster concurrently.

  - *Single Node*: Suitable for small workloads or testing, where only a single worker node is required.

• The version of the *Databricks Runtime* to be used in the cluster; which dictates the version of Spark and individual components such as Python, Scala, and others that get installed.

• The type of virtual machine (VM) used for the worker nodes in the cluster.

• The minimum and maximum number of worker nodes in the cluster.

• The type of VM used for the driver node in the cluster.

• Whether the cluster supports *autoscaling* to dynamically resize the cluster.

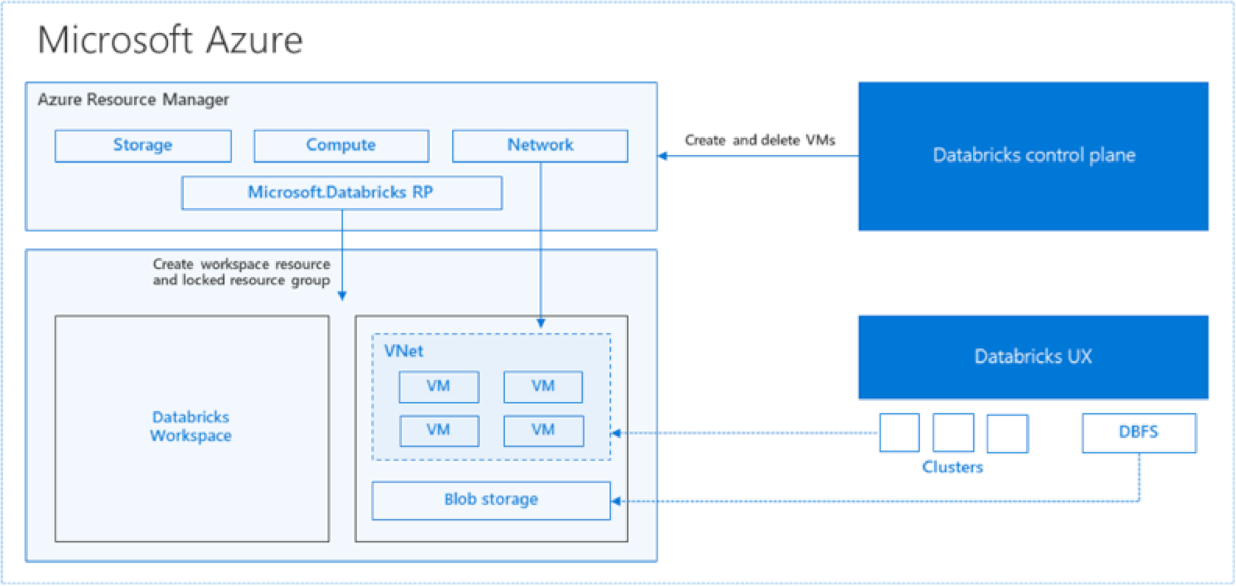
• How long the cluster can remain idle before being shut down automatically.

How Azure manages cluster resources

When you create an Azure Databricks workspace, a *Databricks appliance* is deployed as an Azure resource in your subscription. When you create a cluster in the workspace, you specify the types and sizes of the virtual machines (VMs) to use for both the driver and worker nodes, and some other configuration options, but Azure Databricks manages all other aspects of the cluster.

The Databricks appliance is deployed into Azure as a managed resource group within your subscription. This resource group contains the driver and worker VMs for your clusters, along with other required resources, including a virtual network, a security group, and a storage account. All metadata for your cluster, such as scheduled jobs, is stored in an Azure Database with geo-replication for fault tolerance.

Internally, Azure Kubernetes Service (AKS) is used to run the Azure Databricks control-plane and data-planes via containers running on the latest generation of Azure hardware (Dv3 VMs), with NvMe SSDs capable of blazing 100us latency on high-performance Azure virtual machines with accelerated networking. Azure Databricks utilizes these features of Azure to further improve Spark performance. After the services within your managed resource group are ready, you can manage the Databricks cluster through the Azure Databricks UI and through features such as auto-scaling and auto-termination.



https://learn.microsoft.com/en-us/azure/databricks/clusters/instance-pools/

Question 26: Incorrect

When planning and implementing your Azure Databricks deployments, you have a number of considerations about networking and network security implementation details including which of the following? (Select four)

- ☑

  ACLs

  (Incorrect)

- □

  VNet Peering

  (Correct)

- □

  Azure Private Link

  (Correct)

- □

AAD

- □

Azure VNet service endpoints

(Correct)

- □

Managed Keys

- □

Vault Secrets

- □

TLS

- □

VNet Injection

(Correct)

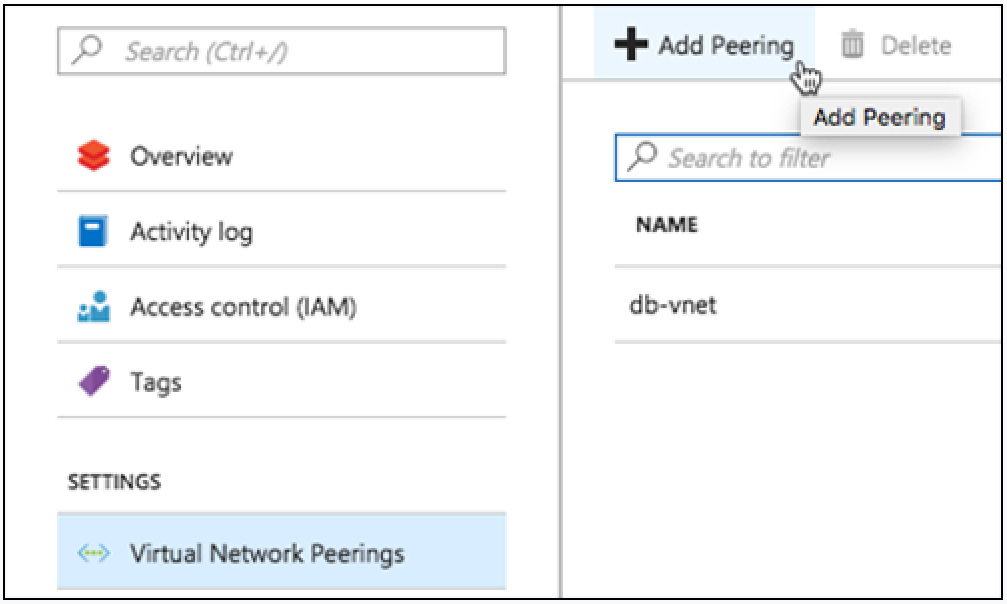
Explanation
When planning and implementing your Azure Databricks deployments, you have a number of considerations about networking and network security implementation details.

Network security

VNet Peering

Virtual network (VNet) peering allows the virtual network in which your Azure Databricks resource is running to peer with another Azure virtual network. Traffic between virtual machines in the peered virtual networks is routed through the Microsoft backbone infrastructure, much like traffic is routed between virtual machines in the same virtual network, through private IP addresses only.

VNet peering is only required if using the standard deployment without VNet injection.



db-vnet

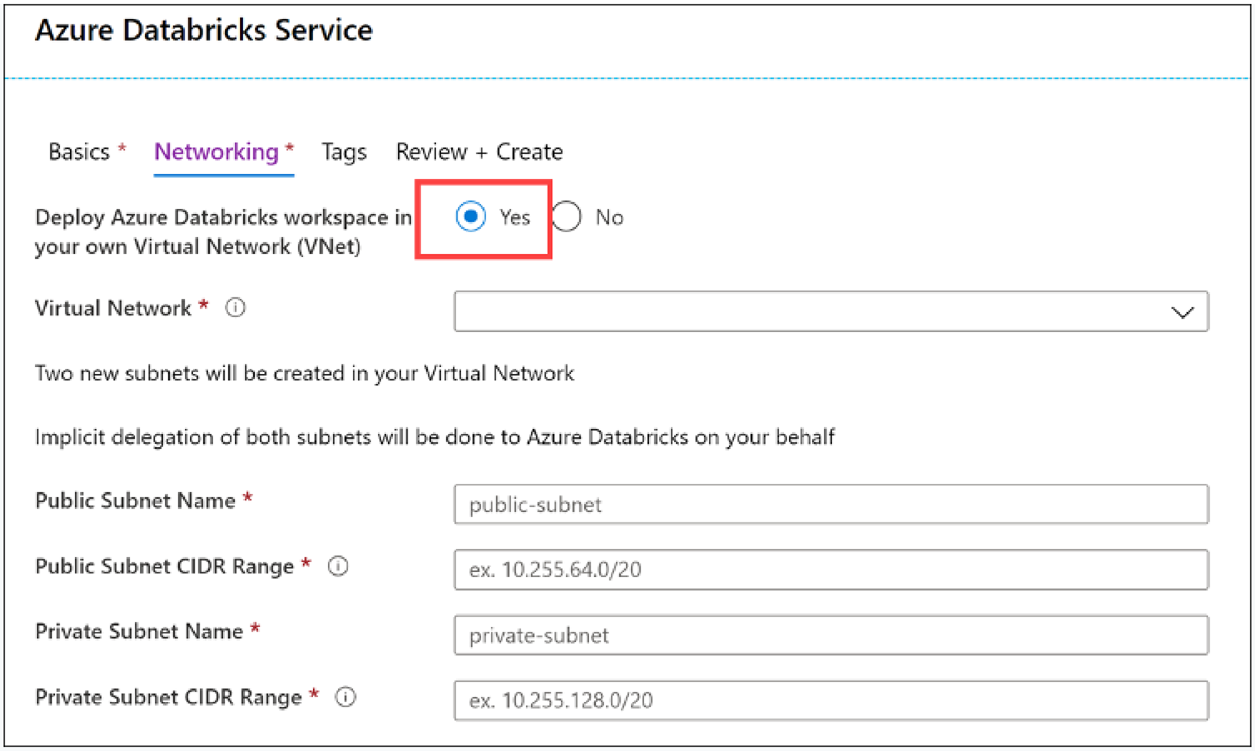
VNet Injection

If you're looking to do specific network customizations, you could deploy Azure Databricks data plane resources in your own VNet. In this scenario, instead of using the managed VNet, which restricts you from making changes, you "bring your own" VNet where you have full control. Azure Databricks will still create the managed VNet, but it will not use it.

Features enabled through VNet injection include:

- On-Premises Data Access
- Single-IP SNAT and Firewall-based filtering via custom routing
- Service Endpoint

To enable VNet injection, select the **Deploy Azure Databricks workspace in your own Virtual Network** option when provisioning your Azure Databricks workspace.


Azure Databricks Service

Basics * Networking * Tags Review + Create

Deploy Azure Databricks workspace in your own Virtual Network (VNet) ● Yes ○ No

Virtual Network *

Two new subnets will be created in your Virtual Network

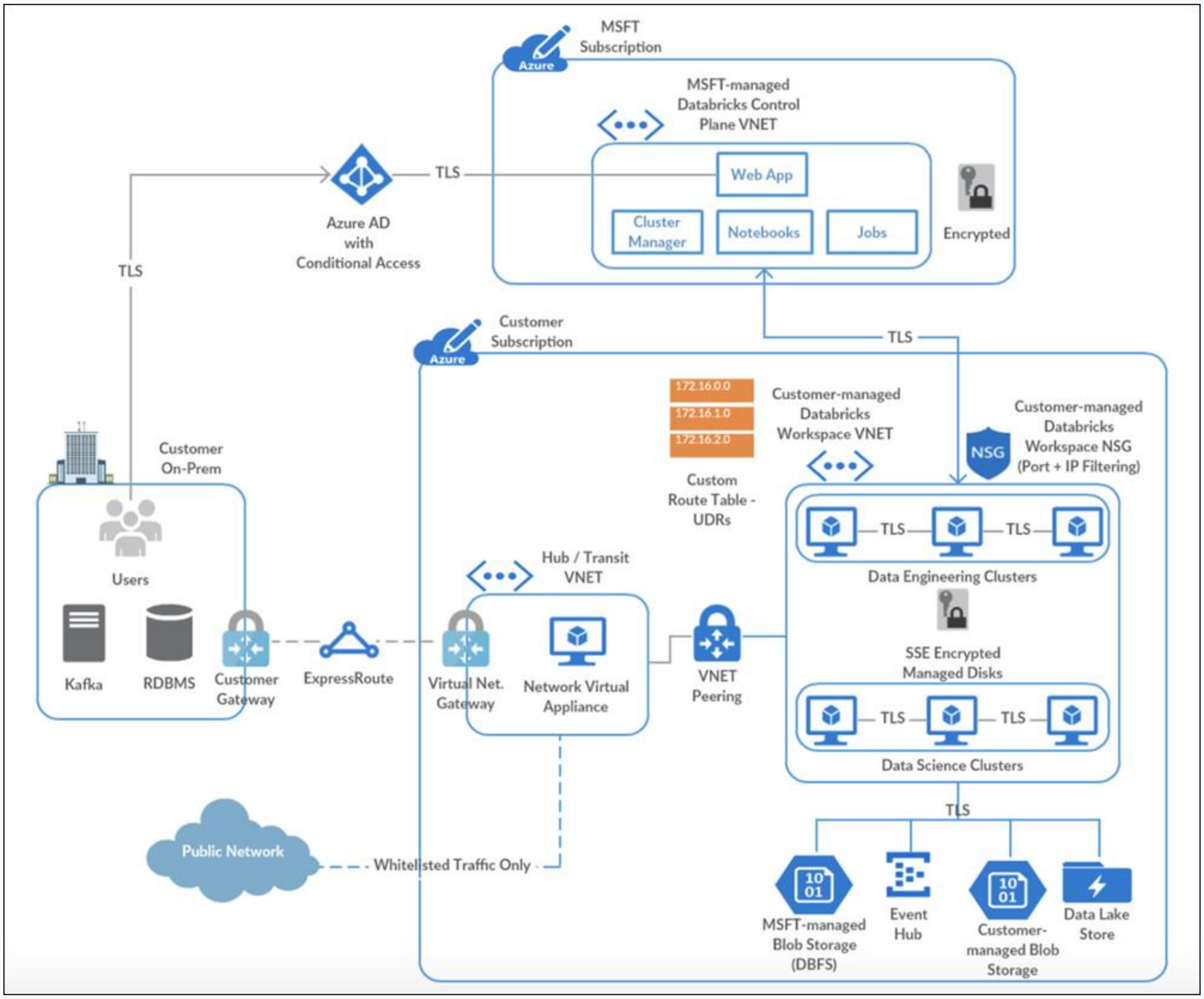
Implicit delegation of both subnets will be done to Azure Databricks on your behalf

Public Subnet Name *	public-subnet
Public Subnet CIDR Range *	ex. 10.255.64.0/20
Private Subnet Name *	private-subnet
Private Subnet CIDR Range *	ex. 10.255.128.0/20

When you compare the deployed Azure Databricks resources in a VNet injection deployment vs. the standard deployment you saw earlier, there are some slight differences. The primary difference is that the clusters in the Data Plane are hosted within a customer-managed Azure Databricks workspace VNet instead of a Microsoft-managed one. The Control Plane is still hosted within a Microsoft-managed VNet, but the TLS connection is still created for you that routes traffic between both VNets. However, the network security groups (NSG) becomes customer-managed as well in this configuration. The only resource in the Data Plane that is still managed by Microsoft is the Blob Storage service that provides DBFS.

Also, inter-node TLS communication between the clusters in the Data Plane is enabled in this deployment. One thing to note is that, while inter-node TLS is more secure, there is a slight impact on performance vs. the non-inter-node TLS found in a basic deployment.

If your Azure Databricks workspace is deployed to your own virtual network (VNet), you can use custom routes, also known as user-defined routes (UDR), to ensure that network traffic is routed correctly for your workspace. For example, if you connect the virtual network to your on-premises network, traffic may be routed through the on-premises network and unable to reach the Azure Databricks control plane. User-defined routes can solve that problem. The diagram below shows UDRs, as well as the other components of a VNet injection deployment.



You can create different Azure Databricks workspaces in the same VNet. However, you will need separate pairs of dedicated subnets per Azure Databricks workspace. As such, the VNet network range has to be fairly large to accommodate those. The VNet CIDR can be anywhere between /16 and /24, and the subnet CIDR can be anywhere between /18 and /26.

Secure connectivity to other Azure data services

Your Azure Databricks deployment likely includes other Azure data services, such as Azure Blob Storage, Azure Data Lake Storage Gen2, Azure Cosmos DB, and Azure Synapse Analytics. We recommend ensuring traffic between Azure Databricks and Azure data services such as these remains on the Azure network backbone, instead of traversing over the public internet. To do this, you should use Azure Private Link or Service Endpoints.

Azure Private Link

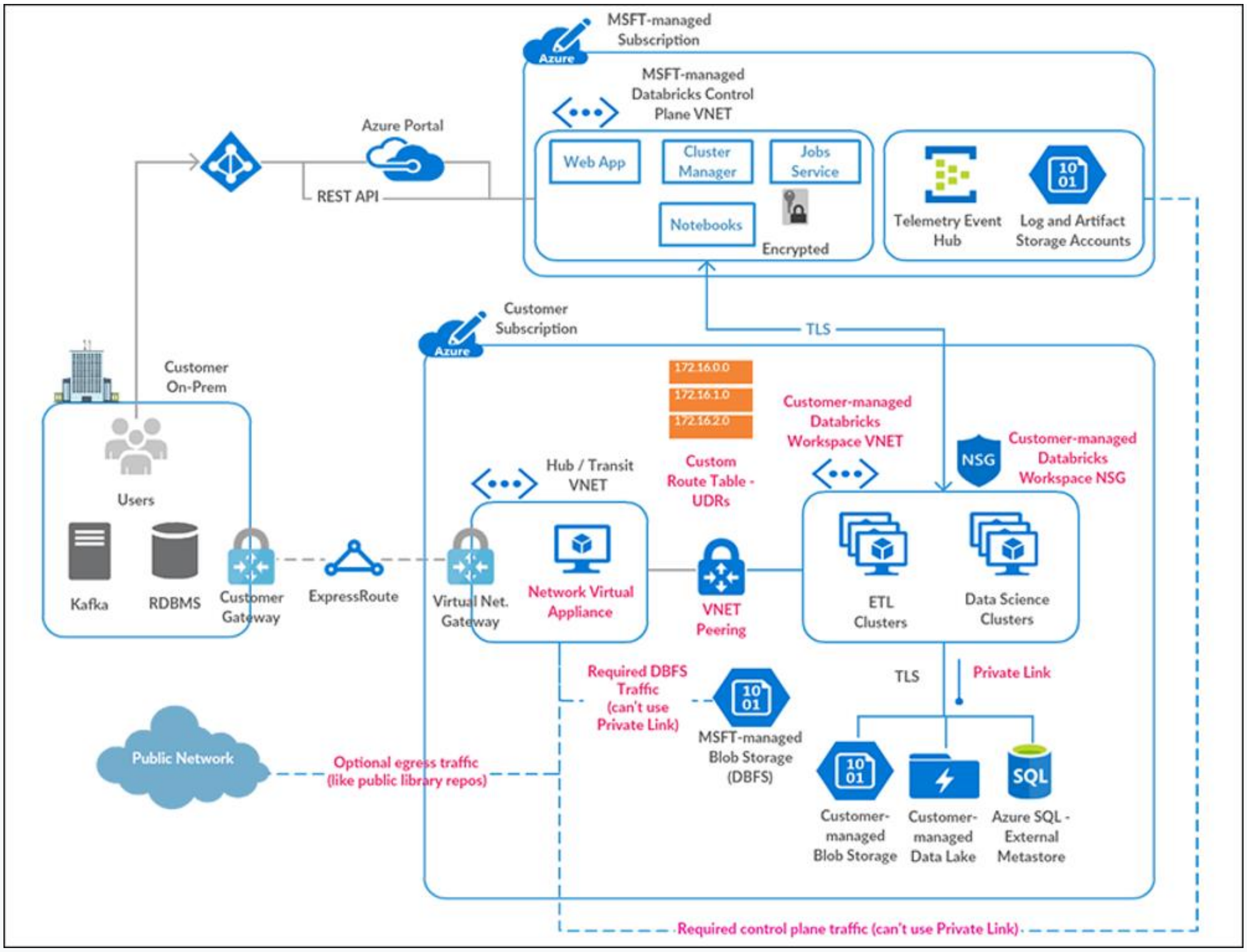
Using Azure Private Link is currently the most secure way to access Azure data services from Azure Databricks. Private Link enables you to access Azure PaaS Services (for example, Azure Storage, Azure Cosmos DB, and SQL Database) and Azure hosted customer/partner services over a Private Endpoint in your virtual network. Traffic between your virtual network and the service traverses over the Microsoft network backbone, eliminating exposure from the public Internet. You can also create your own Private Link Service in your virtual network (VNet) and deliver it privately to your customers.

Azure VNet service endpoints

Virtual Network (VNet) service endpoints extend your virtual network private address space. The endpoints also extend the identity of your VNet to the Azure services over a direct connection. Endpoints allow you to secure your critical Azure service resources to only your virtual networks. Traffic from your VNet to the Azure service always remains on the Microsoft Azure network backbone.

Combining VNet injection and Private Link

The following diagram shows how you may use Private Link in combination with VNet injection in a hub and spoke topology to prevent data exfiltration:


MSFT-managed Subscription
Azure
MSFT-managed Databricks Control Plane VNET
Azure Portal
REST API
Web App
Cluster Manager
Jobs Service
Notebooks
Encrypted
Telemetry Event Hub
Log and Artifact Storage Accounts
10 01
Customer Subscription
Azure
TLS
Customer On-Prem
Users
Kafka
RDBMS
Customer Gateway
ExpressRoute
Virtual Net. Gateway
Hub / Transit VNET
Network Virtual Appliance
172.16.0.0
172.16.1.0
172.16.2.0
Custom Route Table - UDRs
VNET Peering
Customer-managed Databricks Workspace VNET
NSG
Customer-managed Databricks Workspace NSG
ETL Clusters
Data Science Clusters
TLS
Private Link
Required DBFS Traffic (can't use Private Link)
MSFT-managed Blob Storage (DBFS)
Public Network
Optional egress traffic (like public library repos)
Customer-managed Blob Storage
Customer-managed Data Lake
SQL
Azure SQL - External Metastore
Required control plane traffic (can't use Private Link)

Compliance

In many industries, it is imperative to maintain compliance through a combination of following best practices in storing and handling data, and by using services that maintain compliance certifications and attestations.

Azure Databricks has the following compliance certifications:

• HITRUST

• AICPA

• PCI DSS

• ISO 27001

• ISO 27018

- HIPAA (Covered by MSFT Business Associates Agreement (BAA))

- SOC2, Type 2

Audit logs

Databricks provides comprehensive end-to-end audit logs of activities performed by Databricks users, allowing your enterprise to monitor detailed Databricks usage patterns. Azure Monitor integration enables you to capture the audit logs and make then centrally available and fully searchable.

Services / Entities included are:

- Accounts

- Clusters

- DBFS

- Genie

- Jobs

- ACLs

- SSH

- Tables

https://docs.microsoft.com/en-us/azure/security/fundamentals/network-overview

Question 27: **Incorrect**
In some cases, the code-free transformation at scale may not meet your requirements. You can use Azure Data Factory to ingest raw data collected from different sources and work with a range of compute resources such as Azure Databricks.

The following steps are required when implementing data ingestion and transformation using the collective capabilities of Azure Data Factory and Azure Databricks.

The order of the below steps has been shuffled.

a. Perform analysis on data

b. Create Azure storage account

c. Create data workflow pipeline

d. Create an Azure Data Factory

e. Add Databricks notebook to pipeline

Select the correct step order from the below.

- ◉

  c → b → d → e → a

  (Incorrect)

- ○

  b → d → c → e → a

  (Correct)

- ○

  b → c → d → e → a

- ○

  b → d → c → a → e

Explanation

The correct order is: b → d → c → e → a

In some cases, the code-free transformation at scale may not meet your requirements. You can use Azure Data Factory to ingest raw data collected from different sources and work with a range of compute resources such as Azure Databricks, Azure HDInsight, or other compute resources to restructure it as per your requirements.

ADF and Azure Databricks

As an example, the integration of Azure Databricks with ADF allows you to add Databricks notebooks within an ADF pipeline to leverage the analytical and data transformation capabilities of Databricks. You can add a notebook within your data workflow to structure and transform raw data loaded into ADF from different sources. Once the data is transformed using Databricks, you can then load it to any data warehouse source.

Data ingestion and transformation using the collective capabilities of ADF and Azure Databricks essentially involves the following steps:

1. **Create Azure storage account** - The fist step is to create an Azure storage account to store your ingested and transformed data.

2. **Create an Azure Data Factory** - Once you have your storage account setup, you need to create your Azure Data Factory using Azure portal.

3. **Create data workflow pipeline** - After your storage and ADF is up and running, you start by creating a pipeline, where the first step is to copy data from your source using ADF's Copy activity. Copy Activity allows you to copy data from different on-premises and cloud sources.

4. **Add Databricks notebook to pipeline** - Once your data is copied to ADF, you add your Databricks notebook to the pipeline, after copy activity. This notebook may contain syntax and code to transform and clean raw data as required.

5. **Perform analysis on data** - Now that your data is cleaned up and structured into the required format, you can use Databricks notebooks to further train or analyze it to output required results.

https://azure.microsoft.com/en-us/blog/operationalize-azure-databricks-notebooks-using-data-factory/

Question 28: **Incorrect**
Which SparkSQL method reads data from the analytical store?

- ○

  `cosmos.read_db`

- ●

  `cosmos.db`

  (Incorrect)

- ○

  `cosmos.oltp`

- ○

`cosmos.olap`

(Correct)

Explanation
`cosmos.olap` is the method that connects to the analytical store in Azure Cosmos DB.

The syntax to create a Spark table is as follows:

```
SQL
%%sql
-- To select a preferred list of regions in a multi-region Azure Cosmos DB account, add spark.cosmos.preferredRegions '<Region1>,<Region2>' in the config options

create table call_center using cosmos.olap options (
spark.synapse.linkedService '<enter linked service name>',
spark.cosmos.container '<enter container name>'
)
```

https://docs.microsoft.com/en-us/azure/synapse-analytics/synapse-link/how-to-query-analytical-store-spark

Question 29: **Incorrect**

Which technology is typically used as a staging area in a modern data warehousing architecture?

- ○

  Azure Synapse SQL Pools

- ●

  Azure Synapse Spark Lakes

  (Incorrect)

- ○

  Azure Synapse Spark Pools

- ○

  Azure Data Lake

  (Correct)

- ○

Azure Data Pools

- ○

Azure Synapse SQL Lakes

Explanation

Azure Data Lake Store Gen 2 is the technology that will be used to stage data before loading it into the various components of Azure Synapse Analytics.

Azure Data Lake Storage Gen2 is a set of capabilities dedicated to big data analytics, built on Azure Blob storage.

Data Lake Storage Gen2 converges the capabilities of Azure Data Lake Storage Gen1 with Azure Blob storage. For example, Data Lake Storage Gen2 provides file system semantics, file-level security, and scale. Because these capabilities are built on Blob storage, you'll also get low-cost, tiered storage, with high availability/disaster recovery capabilities.

https://docs.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-introduction

Question 30: **Incorrect**

How do you perform `UPSERT` in a Delta dataset?

- ○

Use `MERGE INTO my-table USING data-to-upsert`

(Correct)

- ●

Use `MODIFY my-table UPSERT INTO data-to-upsert`

(Incorrect)

- ○

Use `UPSERT INTO my-table /MERGE`

- ○

Use `UPSERT INTO my-table`

Explanation

Use the `MERGE INTO my-table USING data-to-upsert` syntax to perform `UPSERT` in a Databricks Delta dataset.

https://www.databasejournal.com/features/mssql/using-the-merge-statement-to-perform-an-upsert.html

Question 31: **Incorrect**

Identify the missing word(s) in the following sentence within the context of Microsoft Azure.

Apache Spark is an open-source distributed system that is used for processing big data workloads.

To achieve this capability, … [?]

- ●

  None of the listed options.

  (Incorrect)

- ○

  Spark enables ad hoc data preparation scenarios, where organizations are wanting to unlock insights from their own data stores without going through the formal processes of setting up a data warehouse.

- ○

  Spark automatically adjusts based on your requirements, freeing you up from managing your infrastructure and picking the right size for your solution.

- ○

  Spark performs predictive analytics using both the native features of Azure Synapse Analytics, and integrating with other technologies such as Azure Databricks.

- ○

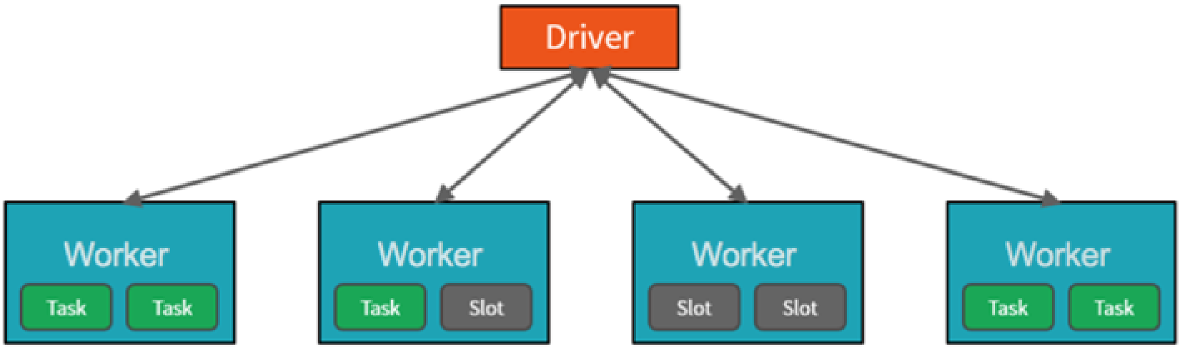
  Spark pools clusters are groups of computers that are treated as a single computer and handle the execution of commands issued from notebooks.

  (Correct)

Explanation

Apache Spark is an open-source distributed system that is used for processing big data workloads. Big data workloads are defined as workloads to handle data that is too large or complex for traditional database systems. Apache Spark processes large amounts of data in memory, which boosts the performance of analyzing big data more effectively, and this capability is available within Azure Synapse Analytics, and is referred to as Spark pools.

To achieve this capability, Spark pools clusters are groups of computers that are treated as a single computer and handle the execution of commands issued from notebooks. The clusters allow processing of data to be parallelized across many computers to improve scale and performance. It consists of a Spark Driver and Worker nodes. The Driver node sends work to the Worker nodes and instructs them to pull data from a specified data source. Moreover, you can configure the number of nodes that are required to perform the task.



Spark pools in Azure Synapse Analytics offer a fully managed Spark service. The benefits of creating a Spark pool in Synapse Analytics include.

Speed and efficiency

Spark instances start in approximately 2 minutes for fewer than 60 nodes and approximately 5 minutes for more than 60 nodes. The instance shuts down, by default, 5 minutes after the last job executed unless it is kept alive by a notebook connection.

Ease of creation

You can create a new Spark pool in Azure Synapse in minutes using the Azure portal, Azure PowerShell, or the Synapse Analytics .NET SDK.

Ease of use

Synapse Analytics includes a custom notebook derived from Nteract. You can use these notebooks for interactive data processing and visualization.

Scalability

Apache Spark in Azure Synapse pools can have Auto-Scale enabled, so that pools scale by adding or removing nodes as needed. Also, Spark pools can be shut down with no loss of data since all the data is stored in Azure Storage or Data Lake Storage.

Support for Azure Data Lake Storage Generation 2

Spark pools in Azure Synapse can use Azure Data Lake Storage Generation 2 as well as BLOB storage.

The primary use case for Apache Spark for Azure Synapse Analytics is to process big data workloads that cannot be handled by Azure Synapse SQL, and where you don’t have an existing Apache Spark implementation.

Perhaps you must perform a complex calculation on large volumes of data. Handling this requirement in Spark pools will be far more efficient than in Synapse SQL. You can pass the data through to the Spark cluster to perform the calculation, and then pass the processed data back into the data warehouse, or back to the data lake.

If you already have a Spark implementation in place already, Azure Synapse Analytics can also integrate with other Spark implementations such as Azure Databricks, so you don’t have to use the feature in Azure Synapse Analytics if you already have a Spark setup already.

Finally, Spark pools in Azure Synapse Analytics come with Anaconda libraries pre-installed. Anaconda provides close to 200 libraries that enables you to use the spark pool to perform machine learning, data analysis, and data visualization. This can enable data scientists and data analysts to interact with the data using the Spark pool too.

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

Question 32: **Incorrect**

See the following code:

```
SQL
COPY INTO dbo.[lineitem] FROM 'https://unsecureaccount.blob.core.windows.net/customerdatasets/folder1/lineitem.csv’
```

What is this code created to do?

- ◉

Move data from a public storage account

(Incorrect)

- ○

  Load data from a private storage account

- ○

  Load data from a public storage account

  (Correct)

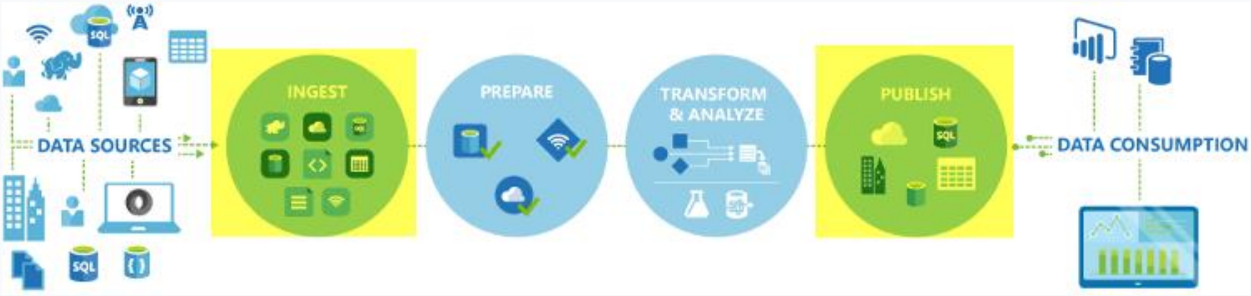
- ○

  Move data from a private storage account

Explanation

The broad capabilities of the Copy Activity allow you to quickly and easily move data into SQL Pools from a variety of sources.

In Azure Data Factory, you can use the Copy activity to copy data among data stores located on-premises and in the cloud. After you copy the data, you can use other activities to further transform and analyze it. You can also use the Copy activity to publish transformation and analysis results for business intelligence (BI) and application consumption.



The Copy activity is executed on an integration runtime. You can use different types of integration runtimes for different data copy scenarios:

• When you're copying data between two data stores that are publicly accessible through the internet from any IP, you can use the Azure integration runtime for the copy activity. This integration runtime is secure, reliable, scalable, and globally available.

• When you're copying data to and from data stores that are located on-premises or in a network with access control (for example, an Azure virtual network), you need to set up a self-hosted integration runtime.

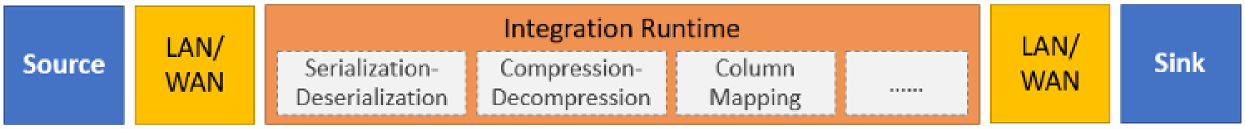
An integration runtime needs to be associated with each source and sink data store. For information about how the Copy activity determines which integration runtime to use, see Determining which IR to use.

To copy data from a source to a sink, the service that runs the Copy activity performs these steps:

1.Reads data from a source data store.

2.Performs serialization/deserialization, compression/decompression, column mapping, and so on. It performs these operations based on the configuration of the input dataset, output dataset, and Copy activity.

3.Writes data to the sink/destination data store.



The Copy Activity supports a large range of data sources and sinks on-premises and in the cloud. It facilitates the efficient, yet flexible parsing and transfer of data or files between systems in an optimized fashion as well as giving you capability of easily converting datasets into other formats.

In the following example, you can load data from a public storage account. Here the `COPY` statement's defaults match the format of the line item csv file.

```
SQL
COPY INTO dbo.[lineitem] FROM
'https://unsecureaccount.blob.core.windows.net/customerdatasets/folder1/lineitem.csv'
```

The default values for csv files of the COPY command are:

• `DATEFORMAT = Session DATEFORMAT`

• `MAXERRORS = 0`

• `COMPRESSION` default is uncompressed

- `FIELDQUOTE = “”`
- `FIELDTERMINATOR = “,”`
- `ROWTERMINATOR = ‘\n’`
- `FIRSTROW = 1`
- `ENCODING = ‘UTF8’`
- `FILE_TYPE = ‘CSV’`
- `IDENTITY_INSERT = ‘OFF’`

https://docs.microsoft.com/en-us/azure/data-factory/copy-activity-overview

Question 33: **Incorrect**

Identify the missing word(s) in the following sentence within the context of Microsoft Azure.

Within an Apache Spark Pool it is possible to configure a fixed size when you disable autoscaling. When you enable autoscale, you can set a minimum and maximum number of nodes in order to control the scale that you'd like. Once you have enabled autoscale, Synapse Analytics will monitor the resource load.

It will continuously monitoring CPU usage, pending memory, free CPU, free memory, and the used memory per node for scaling decisions. It checks these metrics every [?] seconds and makes scaling decisions based on the values.

- ●

  90

  (Incorrect)
- ○

  120
- ○

  45
- ○

60

- ○

  30

  (Correct)

Explanation
Within an Apache Spark Pool it is possible to configure a fixed size when you disable autoscaling. When you enable autoscale, you can set a minimum and maximum number of nodes in order to control the scale that you'd like. Once you have enabled autoscale, Synapse Analytics will monitor for you the requirement of resources of the load. Accordingly, it will scale the number of nodes up or down. There will be continuously monitoring depending on CPU usage, pending memory, free CPU, free memory, and the used memory per node when it comes to the metrics involved to make a decision to scale up or down. **It checks these metrics every 30 seconds and makes scaling decisions based on the values. There's no additional charge for this feature.**

To make it a bit more simple for you, details are below that show the metrics that autoscale enablement of the Spark Pool within Azure Synapse analytics instance checks and collects:

• Total Pending CPU

The total number of cores required to start execution of all pending nodes.

• Total Pending Memory

The total memory (in MB) required to start execution of all pending nodes.

• Total Free CPU

The sum of all unused cores on the active nodes.

• Total Free Memory

The sum of unused memory (in MB) on the active nodes.

• Used Memory per Node

The load on a node. A node on which 10 GB of memory is used, is considered under more load than a worker with 2 GB of used memory.

The metrics will be checked every 30 seconds and the autoscale function will base it decisions of scale-up and scale-down accordingly.

When we look at load-based scale conditions, the autoscale functionality will issue a scale request based on the metrics outlined in the details below:

Scale-up

- Total pending CPU is greater than total free CPU for more than 1 minute.

- Total pending memory is greater than total free memory for more than 1 minute.

Scale-down

- Total pending CPU is less than total free CPU for more than 2 minutes.

- Total pending memory is less than total free memory for more than 2 minutes.

When autoscale, scales up, it will calculate the number of new nodes that would be needed in order to meet the CPU and memory requirements. Next, it will issue the scale-up requests and add the number of nodes required to do the job.

In case autoscale performs the action of scaling down, the decision is based on the number of executors as well as application primaries per node and the CPU and memory requirements. The autoscale functionality will then issue the request to remove a certain number of nodes. What the autoscale functionality will also do, is check which nodes are candidates for removal based on the current job execution. The scale down operation first decommissions the nodes, and then removes them from the cluster.

If you'd like to get started with the autoscale functionality, you'd have to follow the next steps:

Create a serverless Apache Spark pool with Autoscaling

To enable the Autoscale feature, complete the following steps as part of the normal pool creation process:

1.On the **Basics** tab, select the **Enable autoscale** checkbox.

2.Enter the desired values for the following properties:

- **Min** number of nodes.

- **Max** number of nodes.

The initial number of nodes will be the minimum. This value defines the initial size of the instance when it's created. The minimum number of nodes can't be fewer than three.

When we look at the best practices in use for the autoscale feature, consider latency as part of the scale up or down operations. It could take 1 to 5 minutes in order for the scaling operations (whether that's scaling up or down) to complete. Also, when you scale down, the nodes will first be put in decommission state such that there won't be new executors launching on the node. The jobs that are still running, will continue to run and finish, however, the pending jobs will be in a waiting state to be scheduled as normal but with fewer nodes.

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

Question 34: **Correct**

Scenario: You are determining the type of Azure service needed to fit the following specifications and requirements from the available options:

Data classification: Unstructured

Operations:

• Only need to be retrieved by ID.

• Customers require a high number of read operations with low latency.

• Creates and updates will be somewhat infrequent and can have higher latency than read operations.

Latency & throughput: Retrievals by ID need to support low latency and high throughput. Creates and updates can have higher latency than read operations.

Transactional support: Not required

- ●

  Azure Blob Storage

  (Correct)

- ○

  Azure Route Table

- ○

  Azure SQL Database

- ○

  Azure Queue Storage

Explanation

Recommended service: Azure Blob storage

Azure Blob storage supports storing files such as photos and videos. It also works with Azure Content Delivery Network (CDN) by caching the most frequently used content and storing it on edge servers. Azure CDN reduces latency in serving up those images to your users.

By using Azure Blob storage, you can also move images from the hot storage tier to the cool or archive storage tier, to reduce costs and focus throughput on the most frequently viewed images and videos.

https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blobs-introduction

Why not other Azure services?

You could upload your images to Azure App Service, so that the same server that is running your app is serving up your images. This solution would work if you didn't have many files. But if you have lots of files, and a global audience, you'll get more performance results by using Azure Blob storage with Azure CDN.

Question 35: **Incorrect**

Scenario: The Brand Corporation is the science and research branch of the Roxxon Corporation which is managed by Melinda May and Phil Coulson. Melinda and Phil have decided to use Azure for the company to increase its efficiencies and security. Melinda hired you as an advisor to guide many projects to ensure their success.

The IT team is planning to use Spark to process data in files to prepare it for analysis.

Which persona view should you use in the Azure Databricks portal?

- ●

  Machine Learning

(Incorrect)

- ○

  AI

- ○

  Data Science and Engineering

  (Correct)

- ○

  SQL

Explanation

The Data Science and Engineering persona is optimized to help with data engineering tasks such as data processing.

https://learn.microsoft.com/en-us/azure/databricks/scenarios/what-is-azure-databricks-ws

Azure Databricks is a fully managed, cloud-based data analytics platform, which empowers developers to accelerate AI and innovation by simplifying the process of building enterprise-grade data applications. Built as a joint effort by Microsoft and the team that started Apache Spark, Azure Databricks provides data science, engineering, and analytical teams with a single platform for big data processing and machine learning.

By combining the power of Databricks, an end-to-end, managed Apache Spark platform optimized for the cloud, with the enterprise scale and security of Microsoft's Azure platform, Azure Databricks makes it simple to run large-scale Spark workloads.

Identify Azure Databricks workloads

Azure Databricks is a comprehensive platform that offers many data processing capabilities. While you can use the service to support any workload that requires scalable data processing, Azure Databricks is optimized for three specific types of data workload and associated user personas:

- Data Science and Engineering

- Machine Learning

- SQL*

**SQL workloads are only available in premium tier workspaces*.

The Azure Databricks user interface supports three corresponding *persona* views that you can switch between depending on the workload you're implementing.

Get started with Azure Databricks

Azure Databricks is a cloud-based distributed platform for data processing built on Apache Spark. Databricks was designed to unify data science, data engineering, and business data analytics on Spark by creating an easy-to-use environment that enables users to spend more time working effectively with data, and less time focused on managing clusters and infrastructure. As the platform has evolved, it has kept up to date with the latest advances in the Spark runtime and added usability features to support common data workloads in a single, centrally managed interface.

Azure Databricks is hosted on the Microsoft Azure cloud platform and integrated with Azure services such as Azure Active Directory, Azure Storage, Azure Synapse Analytics, and Azure Machine Learning. Organizations can apply their existing capabilities with the Databricks platform and build fully integrated data analytics solutions that work with cloud infrastructure used by other enterprise applications.

Creating an Azure Databricks workspace

To use Azure Databricks, you must create an Azure Databricks *workspace* in your Azure subscription. You can accomplish this by:

- Using the Azure portal user interface.

- Using an Azure Resource Manager (ARM) or Bicep template.

- Using the New-AzDatabricksWorkspace Azure PowerShell cmdlet

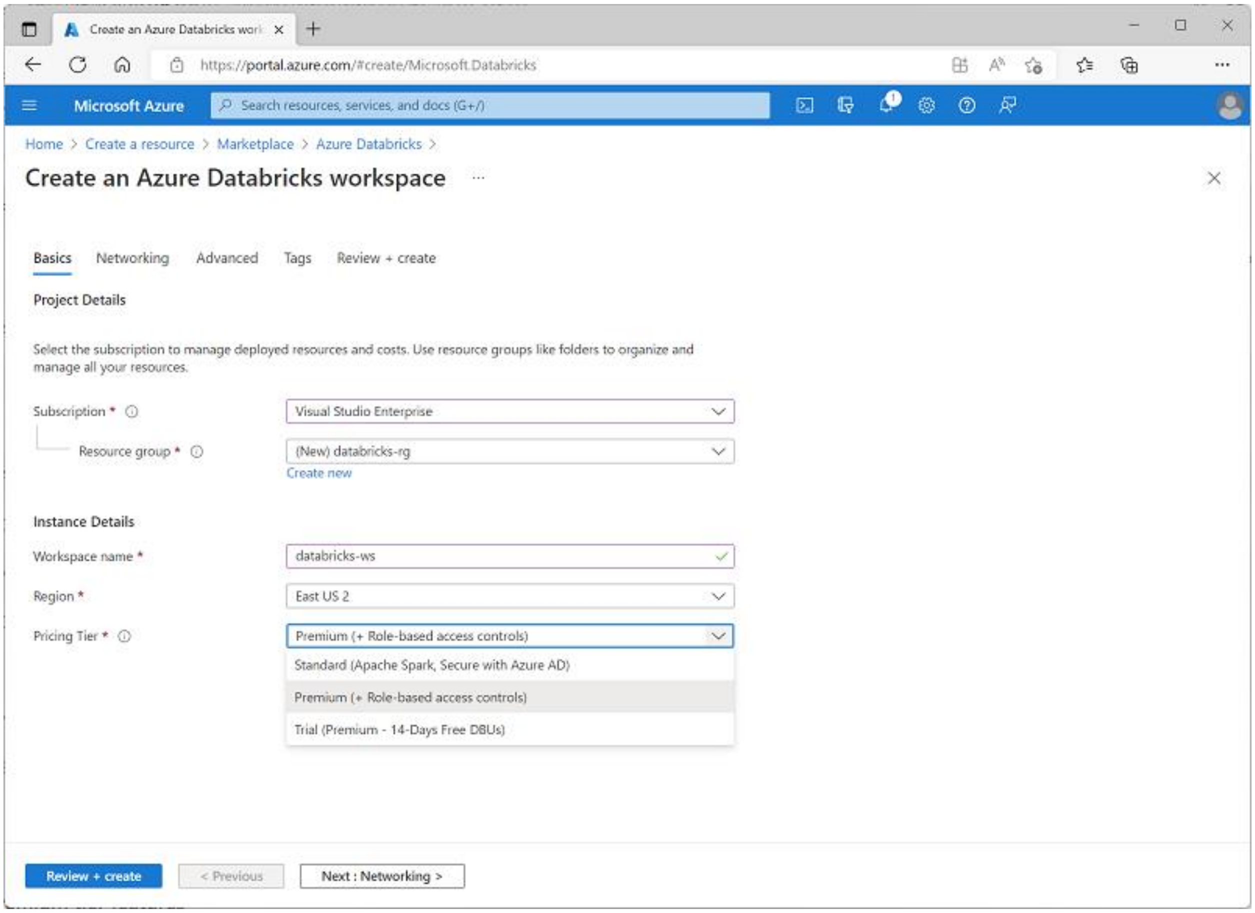
• Using the az databricks workspace create Azure command line interface (CLI) command.

When you create a workspace, you must specify one of the following pricing tiers:

• **Standard** - Core Apache Spark capabilities with Azure AD integration.

• **Premium** - Role-based access controls and other enterprise-level features.

• **Trial** - A 14-day free trial of a premium-level workspace


Create an Azure Databricks worl
https://portal.azure.com/#create/Microsoft.Databricks
Microsoft Azure
Search resources, services, and docs (G+/)
Home > Create a resource > Marketplace > Azure Databricks >
Basics
Networking
Advanced
Tags
Review + create
Review + create
< Previous
Next : Networking >

Create an Azure Databricks workspace

Project Details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *: Visual Studio Enterprise

Resource group *: (New) databricks-rg

Create new

Instance Details

Workspace name *: databricks-ws

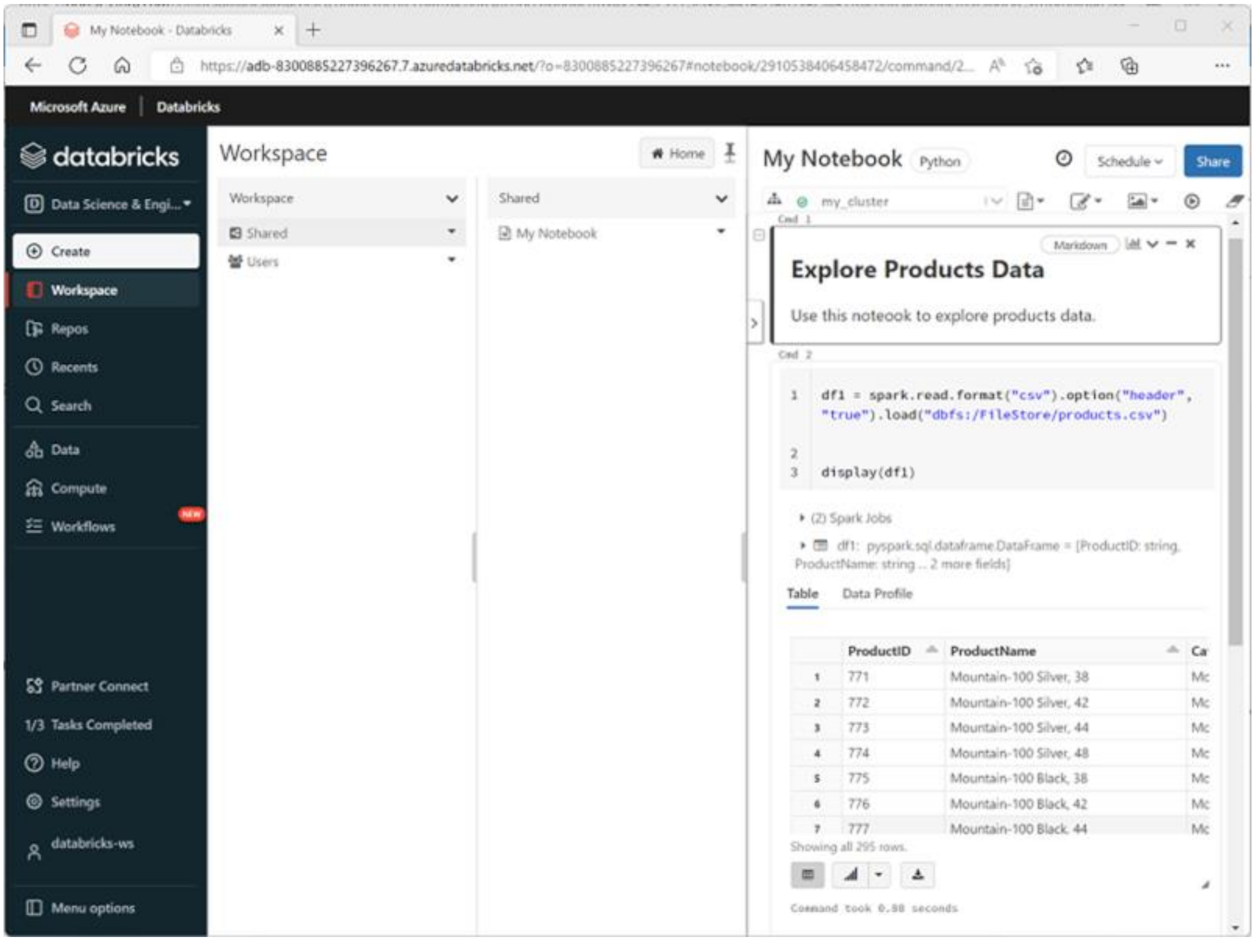
Region *: East US 2

Pricing Tier *: Premium (+ Role-based access controls)

- Standard (Apache Spark, Secure with Azure AD)
- Premium (+ Role-based access controls)
- Trial (Premium - 14-Days Free DBUs)

Using the Azure Databricks portal

After you've provisioned an Azure Databricks workspace, you can use the Azure Databricks portal to work with data and compute resources. The Azure Databricks portal is a web-based user interface through which you can create and manage workspace resources (such as Spark clusters) and work with data in files and tables.



Explore Products Data

Use this noteook to explore products data.

```
df1 = spark.read.format("csv").option("header",
"true").load("dbfs:/FileStore/products.csv")

display(df1)
```

```
▸ (2) Spark Jobs
▸ df1: pyspark.sql.dataframe.DataFrame = [ProductID: string,
ProductName: string ... 2 more fields]
```

	ProductID	ProductName	Ca
1	771	Mountain-100 Silver, 38	Mc
2	772	Mountain-100 Silver, 42	Mc
3	773	Mountain-100 Silver, 44	Mc
4	774	Mountain-100 Silver, 48	Mc
5	775	Mountain-100 Black, 38	Mc
6	776	Mountain-100 Black, 42	Mc
7	777	Mountain-100 Black, 44	Mc

Showing all 295 rows.

```
Command took 0.88 seconds
```

https://learn.microsoft.com/en-us/azure/databricks/introduction/

Question 36: **Correct**

Scenario: You are working as a consultant at Advanced Idea Mechanics (A.I.M.) who is a privately funded think tank organized of a group of brilliant scientists whose sole dedication is to acquire and develop power through technological means. Their goal is to use this power to overthrow the governments of the world. They supply arms and technology to radicals and subversive organizations in order to foster a violent technological revolution of society while making a profit.

The company has 10,000 employees. Most employees are located in Europe. The company supports teams worldwide.

AIM has two main locations: a main office in London, England, and a manufacturing plant in Berlin, Germany.

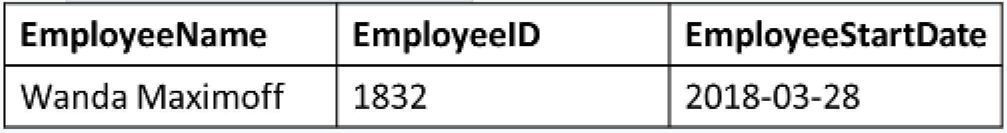
At the moment, you are leading a Workgroup meeting with the IT Team where the topic of discussion is Azure Synapse.

AIM has an Azure Synapse workspace named `aimWorkspace` that contains an Apache Spark database named `aimtestdb`.

The lead developer runs the following command in an Azure Synapse Analytics Spark pool in `aimWorkspace`.

```
CREATE TABLE aimtestdb.aimParquetTable(
EmployeeID int,
EmployeeName string,
EmployeeStartDate date)
```

Using Parquet the developer then employs Spark to insert a row into `aimtestdb.aimParquetTable`. The row contains the following data:



EmployeeName	EmployeeID	EmployeeStartDate
Wanda Maximoff	1832	2018-03-28

Five minutes later, the developer executes the following query from a serverless SQL pool in `aimWorkspace`.

```
SELECT EmployeeID
FROM aimtestdb.dbo.aimParquetTable
WHERE name = 'Wanda Maximoff';
```

What will be returned by the query?

- ◉

  An error

  (Correct)

- ○

  2018-03-28

- ○

Wanda Maximoff

- ○

1832

- ○

A NULL value

Explanation

An error will be thrown because there is a column 'name' in the `WHERE` clause which doesn't exist in the table.

The query should be written as:

```
SELECT EmployeeID
FROM aimtestdb.dbo.aimParquetTable
WHERE employeename = 'Wanda Maximoff';
```

Once a database has been created by a Spark job, you can create tables in it with Spark that use Parquet as the storage format. Table names will be converted to lower case and need to be queried using the lower case name. These tables will immediately become available for querying by any of the Azure Synapse workspace Spark pools. They can also be used from any of the Spark jobs subject to permissions.

Note: For external tables, since they are synchronized to serverless SQL pool asynchronously, there will be a delay until they appear.

Azure Synapse Analytics allows the different workspace computational engines to share databases and Parquet-backed tables between its Apache Spark pools and serverless SQL pool.

Once a database has been created by a Spark job, you can create tables in it with Spark that use Parquet as the storage format. Table names will be converted to lower case and need to be queried using the lower case name. These tables will immediately become available for querying by any of the Azure Synapse workspace Spark pools. They can also be used from any of the Spark jobs subject to permissions.

The Spark created, managed, and external tables are also made available as external tables with the same name in the corresponding synchronized database in serverless SQL pool. Exposing a Spark table in SQL provides more detail on the table synchronization.

Since the tables are synchronized to serverless SQL pool asynchronously, there will be a delay until they appear.

Manage a Spark created table

Use Spark to manage Spark created databases. For example, delete it through a serverless Apache Spark pool job, and create tables in it from Spark.

If you create objects in such a database from serverless SQL pool or try to drop the database, the operation will fail. The original Spark database cannot be changed via serverless SQL pool.

https://docs.microsoft.com/en-us/azure/synapse-analytics/metadata/table

Question 37: **Incorrect**

Scenario: You are the team lead working on a project and a new team member joins your group. He proceeds to review options for an input to an Azure Stream Analytics job your team is working on which requires low latencies and high throughput. He seems uncertain which input he should be using so your ask him *"Which Azure product do you plan to use for the job's input?"*

What should his answer be?

- ●

  Azure Table Storage

  (Incorrect)
- ○

  Azure Data Lake Storage
- ○

  Azure Queue Storage
- ○

  Azure IoT Hub
- ○

  Azure Blob storage
- ○

  Azure Event Hubs

(Correct)

Explanation
Event Hubs consumes data streams from applications at low latencies and high throughput.

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

Question 38: **Incorrect**
What does the CD in CI/CD mean?

- ○

Control Data

- ●

Both Control Data & Continuous Delivery

(Incorrect)

- ○

Continuous Deployment

- ○

Both Continuous Deployment & Continuous Delivery

(Correct)

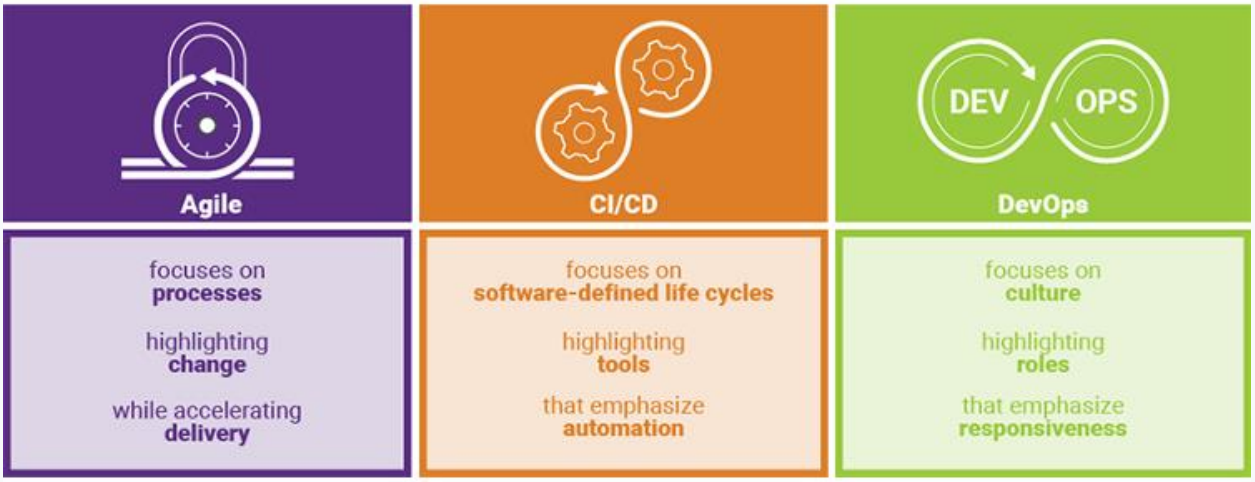
- ○

Continuous Delivery

- ○

Community Development

Explanation
While Agile, CI/CD, and DevOps are different, they support one another. Agile focuses on the development process, CI/CD on practices, and DevOps on culture.


Agile
focuses on processes
highlighting change
while accelerating delivery
CI/CD
focuses on software-defined life cycles
highlighting tools
that emphasize automation
DEV
OPS
DevOps
focuses on culture
highlighting roles
that emphasize responsiveness

• **Agile** focuses on processes highlighting change while accelerating delivery.

• **CI/CD** focuses on software-defined life cycles highlighting tools that emphasize automation.

• **DevOps** focuses on culture highlighting roles that emphasize responsiveness.

https://www.synopsys.com/blogs/software-security/agile-cicd-devops-difference/

Azure DevOps is a collection of services that provide an end-to-end solution for the five core practices of DevOps: planning and tracking, development, build and test, delivery, and monitoring and operations.

It is possible to put an Azure Databricks Notebook under Version Control in an Azure Devops repo. Using Azure DevOps, you can then build Deployment pipelines to manage your release process.

CI/CD with Azure DevOps

Here are some of the features that make it well-suited to CI/CD with Azure Databricks.

• Integrated Git repositories

• Integration with other Azure services

• Automatic virtual machine management for testing builds

- Secure deployment

- Friendly GUI that generates (and accepts) various scripted files

But what is CI/CD?

Continuous Integration

Throughout the development cycle, developers commit code changes locally as they work on new features, bug fixes, etc. If the developers practice continuous integration, they merge their changes back to the main branch as often as possible. Each merge into the master branch triggers a build and automated tests that validate the code changes to ensure successful integration with other incoming changes. This process avoids integration headaches that frequently happen when people wait until the release day before they merge all their changes into the release branch.

Continuous Delivery

Continuous delivery builds on top of continuous integration to ensure you can successfully release new changes in a fast and consistent way. This is because, in addition to the automated builds and testing provided by continuous integration, the release process is automated to the point where you can deploy your application with the click of a button.

Continuous Deployment

Continuous deployment takes continuous delivery a step further by automatically deploying your application without human intervention. This means that merged changes pass through all stages of your production pipeline and, unless any of the tests fail, automatically release to production in a fully automated manner.

Continuous Delivery automates your release process up to the point where human intervention is needed, by clicking a button. Continuous Deployment takes a step further by removing the human intervention and relying on automated tests to automatically determine whether the build should be deployed into production.

Who benefits?

Everyone. Once properly configured, automated testing and deployment can free up your engineering team and enable your data team to push their changes into production. For example:

- Data engineers can easily deploy changes to generate new tables for BI analysts.

- Data scientists can update models being used in production.

• Data analysts can modify scripts being used to generate dashboards.

In short, changes made to a Databricks notebook can be pushed to production with a simple mouse click (and then any amount of oversight that your DevOps team feels is appropriate).

https://docs.microsoft.com/en-us/azure/devops/user-guide/alm-devops-features?view=azure-devops

Question 39: **Incorrect**
What command should be issued to view the list of active streams?

- ○

  Invoke `spark.view.active`

- ●

  Invoke `spark.streams.show`

  (Incorrect)

- ○

  Invoke `spark.streams.active`

  (Correct)

- ○

  Invoke `spark.view.activeStreams`

Explanation
Invoke `spark.streams.active` which is the correct syntax to view the list of active streams.

https://spark.apache.org/docs/latest/structured-streaming-programming-guide.html

Question 40: **Incorrect**
Azure Storage provides a REST API to work with the containers and data stored in each account. To work with data in a storage account, your app will need which pieces of data? (Select two)

- □

Access key

(Correct)

- ☑

REST API endpoint

(Correct)

- □

Instance key

- □

Private access key

- □

Subscription key

- □

Public access key

Explanation

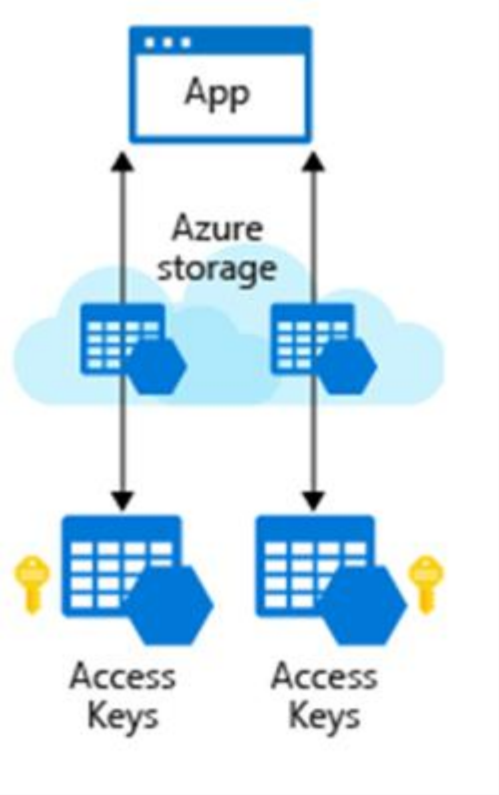
Azure Storage provides a REST API to work with the containers and data stored in each account. To work with data in a storage account, your app will need two pieces of data:

• Access key

• REST API endpoint

Security access keys

Each storage account has two unique *access keys* that are used to secure the storage account. If your app needs to connect to multiple storage accounts, your app will require an access key for each storage account.



REST API endpoint

In addition to access keys for authentication to storage accounts, your app will need to know the storage service endpoints to issue the REST requests.

The REST endpoint is a combination of your storage account *name*, the data type, and a known domain. For example:

Data type: Blobs

Example endpoint: https://[name].blob.core.windows.net/

Data type: Queues

Example endpoint: https://[name].queue.core.windows.net/

Data type: Table

Example endpoint: https://[name].table.core.windows.net/

Data type: Files

Example endpoint: https://[name].file.core.windows.net/

If you have a custom domain tied to Azure, then you can also create a custom domain URL for the endpoint.

https://docs.microsoft.com/en-us/rest/api/storageservices/blob-service-rest-api

Question 41: **Incorrect**
Blob storage is optimized for storing massive amounts of unstructured data.

What does unstructured mean?

- ○

  Blobs can't contain structured data, like JSON or XML.

- ●

  Blobs can't be organized or named.

  (Incorrect)

- ○

  There are no restrictions on the type of data you can store in blobs.

  (Correct)

- ○

  None of the listed options.

Explanation
Azure Blob storage is Microsoft's object storage solution for the cloud. Blob storage is optimized for storing massive amounts of unstructured data. Unstructured data is data that doesn't adhere to a particular data model or definition, such as text or binary data.

https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blobs-introduction

Question 42: **Incorrect**
Knowing now the different concepts of spark it is imperative to understand how it fits in with the different Data services on Azure.

Which of the following is best described by:

"An implementation by Microsoft of Open Source Spark, managed on the Azure Platform. You can use this for a spark environment when you are aware of the benefits of Apache

Spark in its OSS form, but you want an SLA. Usually this is of interest to Open Source Professionals needing an SLA as well as Data Platform experts experienced with Microsoft.”

- ●

  Apache Spark

  (Incorrect)

- ○

  HDI

  (Correct)

- ○

  Spark Pools in Azure Synapse Analytics

- ○

  Azure Databricks

Explanation

There are two concepts within Apache Spark Pools in Azure Synapse Analytics, namely Spark pools and Spark Instances. In short, they do the following:

Spark Pools:

• Exists as Metadata

• Creates a Spark Instance

• No costs associated with creating Pool

• Permissions can be applied

• Best practices

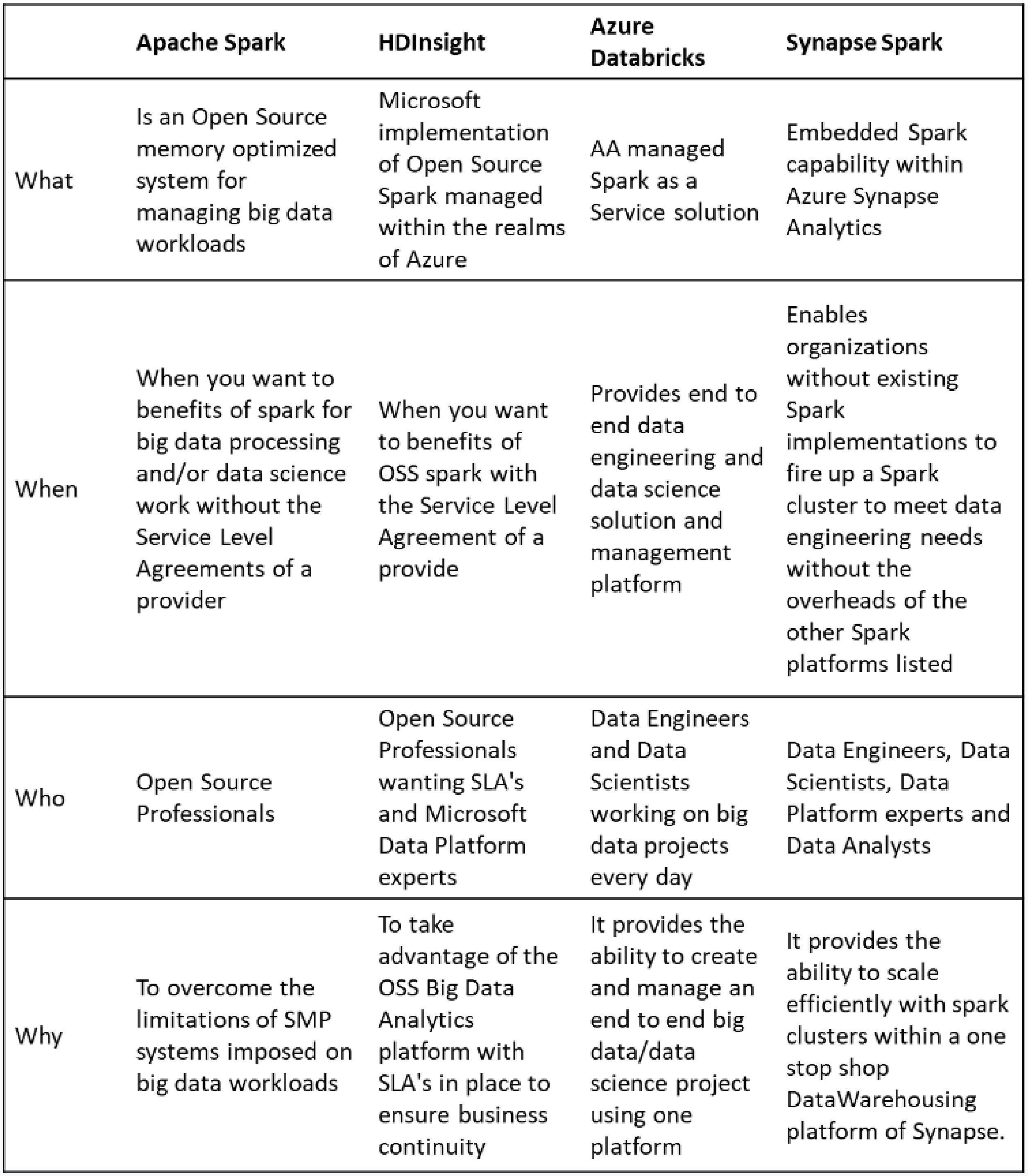
Spark Instances:

• Created when connected to Spark Pool, Session, or Job

• Multiple users can have access

- Reusable

Knowing now the different concepts of spark it is imperative to understand how it fits in with the different Data services on Azure. Below is a table where "the when to use what" is outlined:



	Apache Spark	HDInsight	Azure Databricks	Synapse Spark
What	Is an Open Source memory optimized system for managing big data workloads	Microsoft implementation of Open Source Spark managed within the realms of Azure	AA managed Spark as a Service solution	Embedded Spark capability within Azure Synapse Analytics
When	When you want to benefits of spark for big data processing and/or data science work without the Service Level Agreements of a provider	When you want to benefits of OSS spark with the Service Level Agreement of a provide	Provides end to end data engineering and data science solution and management platform	Enables organizations without existing Spark implementations to fire up a Spark cluster to meet data engineering needs without the overheads of the other Spark platforms listed
Who	Open Source Professionals	Open Source Professionals wanting SLA's and Microsoft Data Platform experts	Data Engineers and Data Scientists working on big data projects every day	Data Engineers, Data Scientists, Data Platform experts and Data Analysts
Why	To overcome the limitations of SMP systems imposed on big data workloads	To take advantage of the OSS Big Data Analytics platform with SLA's in place to ensure business continuity	It provides the ability to create and manage an end to end big data/data science project using one platform	It provides the ability to scale efficiently with spark clusters within a one stop shop DataWarehousing platform of Synapse.

***Spark Pools in Azure Synapse Analytics*:** Spark in Azure Synapse Analytics is a capability of Spark embedded in Azure Synapse Analytics in which organizations that don't have existing spark implementations yet, get the functionality to spin up a spark cluster to meet data engineering needs without the overhead of the other Spark Platforms listed. Data Engineers, Data scientist, Data Platform Experts, and Data Analyst can come together within Synapse Analytics where the Spark cluster is spun up quickly to meet the needs. It provides scale in an efficient way for Spark Clusters and integrates with the one stop shop Data warehousing platform of Synapse.

***Apache Spark*:** Apache Spark is an open-source memory optimized system for managing big data workloads, which is used when you want a spark engine for big data processing or data science where you don't mind that there is no SLA provided. Usually it is of interest of Open Source Professionals and the reason for Apache spark is to overcome the limitations of what was known as SMP systems for big data workloads.

***HDI*:** HDI is an implementation by Microsoft of Open Source Spark, managed on the Azure Platform. You can use HDI for a spark environment when you are aware of the benefits of Apache Spark in its OSS form, but you want a SLA. Usually this of interest of Open Source Professionals needing an SLA as well as Data Platform experts experienced with Microsoft.

***Azure Databricks*:** Azure Databricks is a managed Spark as a Service propriety Solution that provides an end to end data engineering/data science platform as a solution. Azure Databricks is of interest for Data Engineers and Data Scientists, working on big data projects daily because it provides the whole platform in which you have the ability to create and manage the big data/data science pipelines/projects all on one platform.

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

Question 43: **Incorrect**

Identify the missing word(s) in the following sentence within the context of Microsoft Azure.

Scenario: You have created a storage account name using a standardized naming convention within your department.

Your teammate is concerned with this practice because the name of a storage account must be [?].

- ●

  None of the listed options

  (Incorrect)

- ○

  Unique within the containing resource group

- ○

  Globally unique

  (Correct)

- ○

  Unique within your Azure subscription

Explanation

The storage account name is used as part of the URI for API access, so it must be globally unique.

https://docs.microsoft.com/en-us/powershell/module/servicemanagement/azure.service/new-azurestorageaccount?view=azuresmps-4.0.0

Question 44: **Incorrect**

Identify the missing word(s) in the following sentence within the context of Microsoft Azure.

Security and infrastructure configuration go hand-in-hand. When you set up your Azure Databricks workspace(s) and related services, you need to make sure that security considerations do not take a back seat during the architecture design.

With regards to the Workspace to VNet ratio, Microsoft recommends … [?]

- ●

  that you should deploy between two and fifteen workspaces in any VNet.

  (Incorrect)

- ○

  that you should deploy at least two workspaces in any VNet.

- ○

  that you should deploy a maximum of ten workspaces in any VNet.

- ○

  that you should only deploy one workspace in any VNet.

(Correct)

Explanation

Security and infrastructure configuration go hand-in-hand. When you set up your Azure Databricks workspace(s) and related services, you need to make sure that security considerations do not take a back seat during the architecture design.

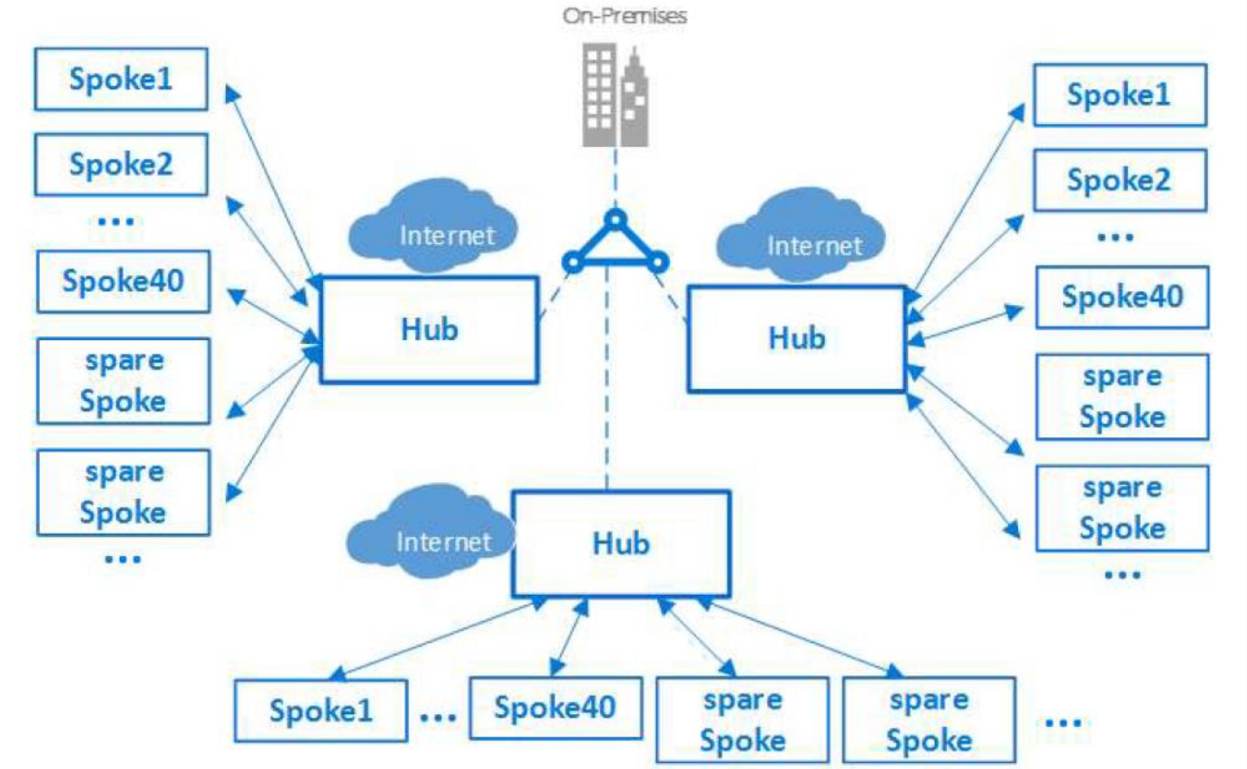
Consider isolating each workspace in its own VNet

While you can deploy more than one Workspace in a VNet by keeping the associated subnet pairs separate from other workspaces, MS recommends that you should only deploy one workspace in any VNet. Doing this perfectly aligns with the ADB's Workspace level isolation model. Most often organizations consider putting multiple workspaces in the same VNet so that they all can share some common networking resource, like DNS, also placed in the same VNet because the private address space in a VNet is shared by all resources. You can easily achieve the same while keeping the Workspaces separate by following the hub and spoke model and using VNet Peering to extend the private IP space of the workspace VNet. Here are the steps:

1.Deploy each Workspace in its own spoke VNet.

2.Put all the common networking resources in a central hub VNet, such as your custom DNS server.

3.Join the Workspace spokes with the central networking hub using VNet Peering



Do not store any production data in Default Databricks Filesystem (DBFS) Folders

This recommendation is driven by security and data availability concerns. Every Workspace comes with a default Databricks File System (DBFS), primarily designed to store libraries and other system-level configuration artifacts such as initialization scripts. You should not store any production data in it, because:

1. The lifecycle of default DBFS is tied to the Workspace. Deleting the workspace will also delete the default DBFS and permanently remove its contents.

2. One can't restrict access to this default folder and its contents.

Important: This recommendation doesn't apply to Blob or ADLS folders explicitly mounted as DBFS by the end user.

Always hide secrets in a key vault

It is a significant security risk to expose sensitive data such as access credentials openly in Notebooks or other places such as job configs, initialization scripts, etc. You should always use a vault to securely store and access them. You can either use ADB's internal Key Vault for this purpose or use Azure's Key Vault (AKV) service.

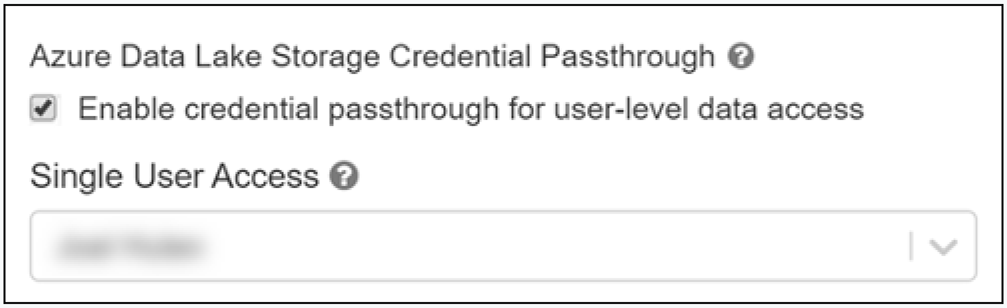
If using Azure Key Vault, create separate AKV-backed secret scopes and corresponding AKVs to store credentials pertaining to different data stores. This will help prevent users from accessing credentials that they might not have access to. Since access controls are applicable to the entire secret scope, users with access to the scope will see all secrets for the AKV associated with that scope.

Access control - Azure Data Lake Storage (ADLS) passthrough

When enabled, authentication automatically takes place in Azure Data Lake Storage (ADLS) from Azure Databricks clusters using the same Azure Active Directory (Azure AD) identity that one uses to log into Azure Databricks. Commands running on a configured cluster will be able to read and write data in ADLS without needing to configure service principal credentials. Any ACLs applied at the folder or file level in ADLS are enforced based on the user's identity.

ADLS Passthrough is configured when you create a cluster in the Azure Databricks workspace. ADLS Gen1 requires Databricks Runtime 5.1+. ADLS Gen2 requires 5.3+.

On a *standard cluster*, when you enable this setting you must set single user access to one of the Azure Active Directory (AAD) users in the Azure Databricks workspace. [Only one user is allowed to run commands](#) on this cluster when Credential Passthrough is enabled.



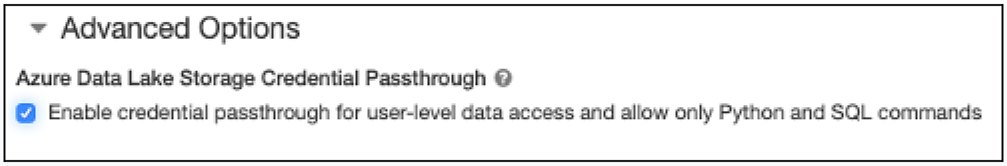
Azure Data Lake Storage Credential Passthrough

☑ Enable credential passthrough for user-level data access

Single User Access

[redacted]

High-concurrency clusters can be shared by multiple users. When you enable ADLS Passthrough on this type of cluster, it does not require you to select a single user.



Azure Data Lake Storage Credential Passthrough

☑ Enable credential passthrough for user-level data access and allow only Python and SQL commands

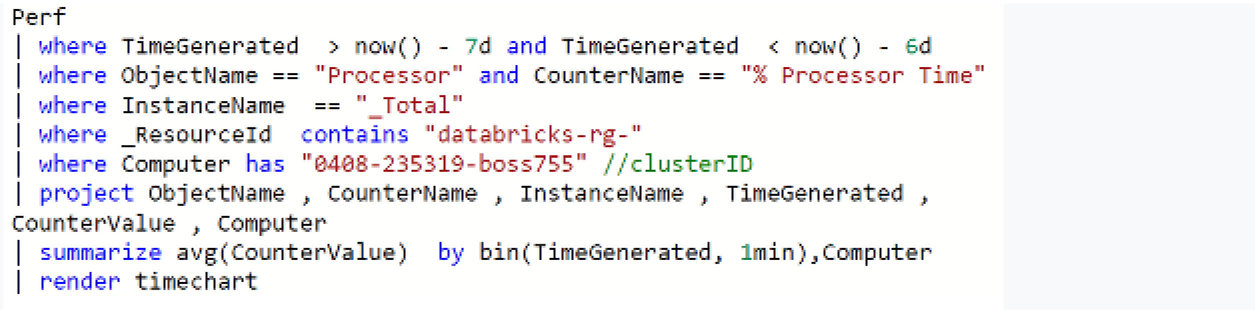
Configure audit logs and resource utilization metrics to monitor activity

An important facet of monitoring is understanding the resource utilization in Azure Databricks clusters. You can also extend this to understanding utilization across all clusters in a workspace. This information is useful in arriving at the correct cluster and VM sizes. Each VM does have a set of limits (cores/disk throughput/network throughput) which play an important role in determining the performance profile of an Azure Databricks job.

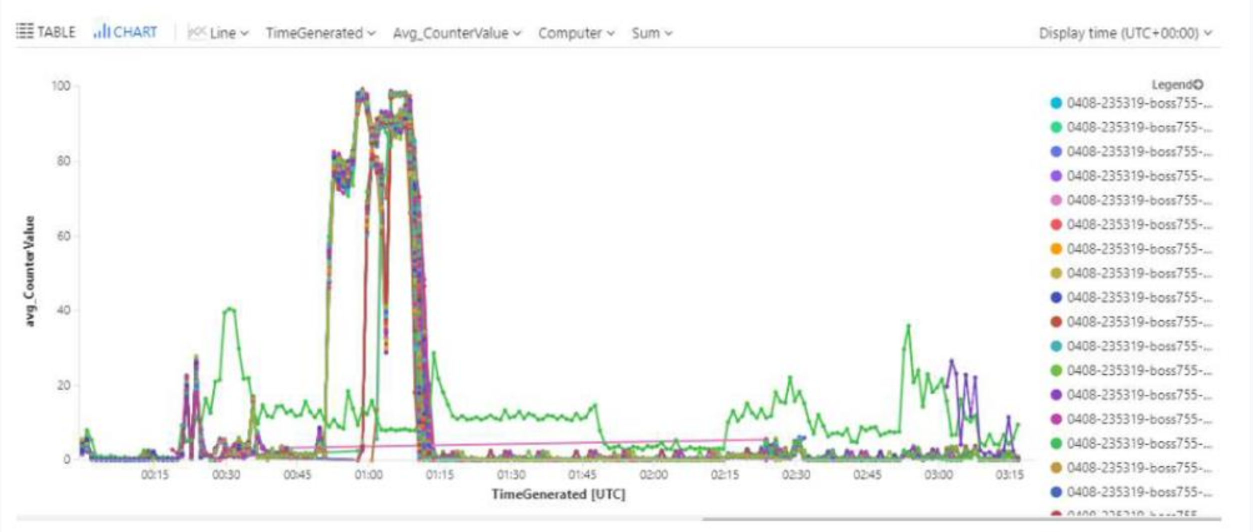
In order to get utilization metrics of an Azure Databricks cluster, you can stream the VM's metrics to an Azure Log Analytics Workspace (see Appendix A) by installing the Log Analytics Agent on each cluster node.

Querying VM metrics in Log Analytics once you have started the collection using the above document

You can use Log analytics directly to query the Perf data. Here is an example of a query which charts out CPU for the VMs in question for a specific cluster ID. See log analytics overview for further documentation on log analytics and query syntax.



```
Perf
| where TimeGenerated  > now() - 7d and TimeGenerated  < now() - 6d
| where ObjectName == "Processor" and CounterName == "% Processor Time"
| where InstanceName  == "_Total"
| where _ResourceId  contains "databricks-rg-"
| where Computer has "0408-235319-boss755" //clusterID
| project ObjectName , CounterName , InstanceName , TimeGenerated ,
CounterValue , Computer
| summarize avg(CounterValue)  by bin(TimeGenerated, 1min),Computer
| render timechart
```



References

1.https://docs.microsoft.com/azure/azure-monitor/learn/quick-collect-linux-computer

2. https://github.com/Microsoft/OMS-Agent-for-Linux/blob/master/docs/OMS-Agent-for-Linux.md

3. https://github.com/Microsoft/OMS-Agent-for-Linux/blob/master/docs/Troubleshooting.md

Question 45: **Incorrect**
Identify the missing word(s) in the following sentence within the context of Microsoft Azure.

Azure Synapse Analytics can work by acting as the one stop shop to meet all of your analytical needs in an integrated environment.

[?] offers both serverless and dedicated resource models to work with both descriptive and diagnostic analytical scenarios. This is a distributed query system that enables you to implement data warehousing and data virtualization scenarios using standard T-SQL.

- ◉

Azure Synapse Pipelines

(Incorrect)

- ○

  Azure Cosmos DB

- ○

  Apache Spark for Azure Synapse

- ○

  Azure Synapse SQL

  (Correct)

- ○

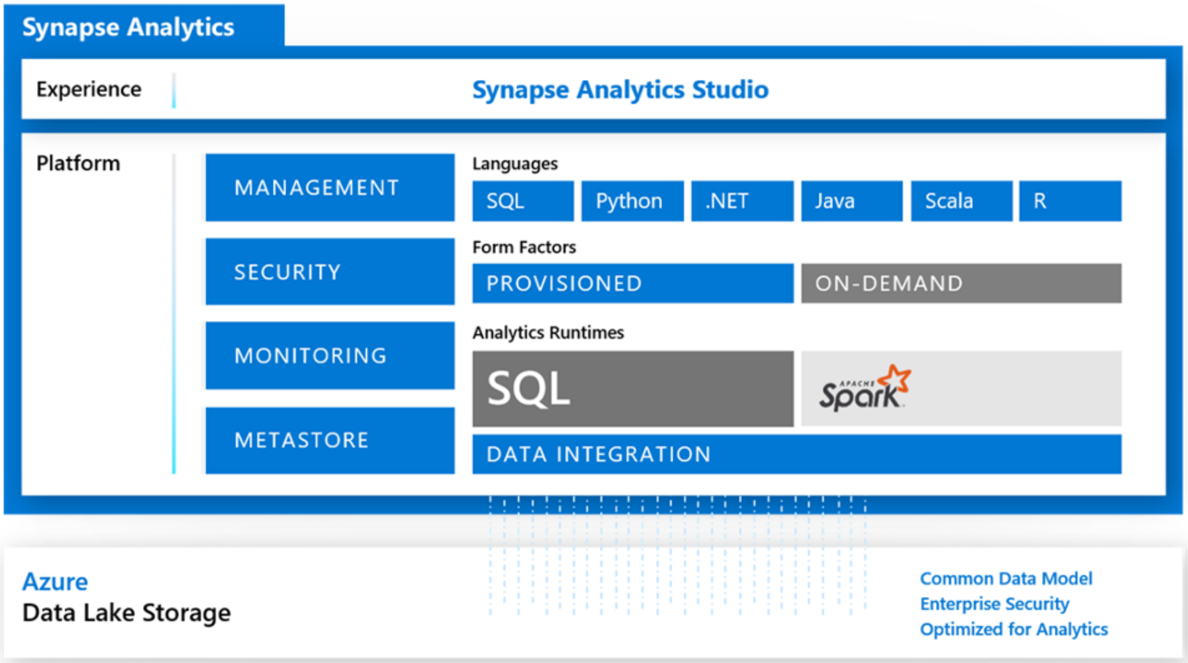
  Azure Synapse Link

Explanation

Azure Synapse Analytics can work by acting as the one stop shop to meet all of your analytical needs in an integrated environment. It does this by providing the following capabilities:

Analytics capabilities offered through Azure Synapse SQL through either dedicated SQL pools or SQL Serverless pools

Azure Synapse SQL is a distributed query system that enables you to implement data warehousing and data virtualization scenarios using standard T-SQL experiences familiar to data engineers. Synapse SQL offers both serverless and dedicated resource models to work with both descriptive and diagnostic analytical scenarios. For predictable performance and cost, create dedicated SQL pools to reserve processing power for data stored in SQL tables. For unplanned or ad-hoc workloads, use the always-available, serverless SQL endpoint.



Apache Spark pool with full support for Scala, Python, SparkSQL, and C#

You can develop big data engineering and machine learning solutions using Apache Spark for Azure Synapse. You can take advantage of the big data computation engine to deal with complex compute transformations that would take too long in a data warehouse. For machine learning workloads, you can use SparkML algorithms and AzureML integration for Apache Spark 2.4 with built-in support for Linux Foundation Delta Lake. There is a simple model for provisioning and scaling the Spark clusters to meet your compute needs, regardless of the operations that you are performing on the data.

Data integration to integrate your data with Azure Synapse Pipelines

Azure Synapse Pipelines leverages the capabilities of Azure Data Factory and is the cloud-based ETL and data integration service that allows you to create data-driven workflows for orchestrating data movement and transforming data at scale. Using Azure Synapse Pipelines, you can create and schedule data-driven workflows (called pipelines) that can ingest data from disparate data stores. You can build complex ETL processes that transform data visually with data flows or by using compute services such as Azure HDInsight Hadoop, or Azure Databricks.

Perform operational analytics with near real-time hybrid transactional and analytical processing with Azure Synapse Link

Azure Synapse Analytics enables you to reach out to operational data using Azure Synapse Link, and is achieved without impacting the performance of the transactional data store. For this to happen, you have to enable the feature within both Azure Synapse Analytics, and within the data store to which Azure Synapse Analytics will connect, such as Azure Cosmos DB. In the case of Azure Cosmos DB, this will create an analytical data store. As data changes in the transactional system, the changed data is fed to the analytical store in a Column store format from which Azure Synapse Link can query with no disruption to the source system.

https://docs.microsoft.com/en-us/azure/synapse-analytics/overview-what-is

Question 46: **Incorrect**

Most Azure Data Factory users develop using the user experience. Azure Data Factory is available in a variety of software development kits (SDKs) for anyone who wish to develop programmatically.

Which of the following allow programmatic interaction with Azure Data Factory?

- ☑

  JavaScript

  (Incorrect)

- ☐

  C++

- ☐

  REST APIs

  (Correct)

- ☐

  Java

- ☐

  PowerShell

  (Correct)

- □

  ARM Templates

  (Correct)

- □

  .NET

  (Correct)

- □

  Python

  (Correct)

Explanation
While most Azure Data Factory users develop using the user experience, Azure Data Factory is available in a variety of software development kits (SDKs) for anyone who wish to develop programmatically. When using an SDK, a user works directly against the Azure Data Factory service and all updates are immediately applied to the factory.

It is possible to interact programmatically with Azure Data Factory using the following:

• Python

• .NET

• REST APIs

• PowerShell

• Azure Resource Manager Templates

• Data flow scripts

Data flow script (DFS) is the underlying metadata, similar to a coding language, that is used to execute the transformations that are included in a mapping data flow. Every transformation is represented by a series of properties that provide the necessary information to run the job properly. The script is visible and editable from ADF by clicking on the "script" button on the top ribbon of the browser UI.

https://docs.microsoft.com/en-us/azure/data-factory/monitor-programmatically

Question 47: **Incorrect**
Large data projects can be complex. The projects often involve hundreds of decisions. Multiple people are typically involved, and each person helps take the project from design to production.

Roles such as business stakeholders, business analysts, and business intelligence developers are well known and valuable.

Which of the available roles is best described by the following:

“Provisions and sets up data platform technologies that are on-premises and in the cloud. They manage and secure the flow of structured and unstructured data from multiple sources. The data platforms they use can include relational databases, nonrelational databases, data streams, and file stores. They ensure that data services securely and seamlessly integrate with other data platform technologies or application services such as Azure Cognitive Services, Azure Search, or even bots.”

- ○

  Project Manager

- ○

  Solution Architects

- ○

  AI Engineer

- ●

  Data Scientist

  (Incorrect)

- ○

  RPA Developers

- ○

  Data Engineer

(Correct)

- ○

  System Administrators

- ○

  BI Engineer

Explanation

Data Engineer

Data engineers primarily provision data stores. They make sure that massive amounts of data are securely and cost-effectively extracted, loaded, and transformed.

Data engineers provision and set up data platform technologies that are on-premises and in the cloud. They manage and secure the flow of structured and unstructured data from multiple sources. The data platforms they use can include relational databases, nonrelational databases, data streams, and file stores. Data engineers also ensure that data services securely and seamlessly integrate with other data platform technologies or application services such as Azure Cognitive Services, Azure Search, or even bots.

The Azure data engineer focuses on data-related tasks in Azure. Primary responsibilities include using services and tools to ingest, egress, and transform data from multiple sources. Azure data engineers collaborate with business stakeholders to identify and meet data requirements. They design and implement solutions. They also manage, monitor, and ensure the security and privacy of data to satisfy business needs.

The role of data engineer is different from the role of a database administrator. A data engineer's scope of work goes well beyond looking after a database and the server where it's hosted. Data engineers must also get, ingest, transform, validate, and clean up data to meet business requirements.

Both database administrators and business intelligence professionals can easily transition to a data engineer role. They just need to learn the tools and technology that are used to process large amounts of data.

https://www.whizlabs.com/blog/azure-data-engineer-roles/

Question 48: **Incorrect**

Identify the missing word(s) in the following sentence within the context of Microsoft Azure.

Azure Cosmos DB analytical store is a fully isolated column store for enabling large-scale analytics against operational data in your Azure Cosmos DB, without any impact to your transactional workloads.

There are two schema representation modes for data stored in the analytical store.

• For SQL (Core) API accounts, when analytical store is enabled, the default schema representation in the analytical store is [A].

• For Azure Cosmos DB API for MongoDB accounts, the default schema representation in the analytical store is [B].

- ●

  [A] Static schema representation, [B] Dynamic schema representation

  (Incorrect)

- ○

  [A] Dynamic schema representation, [B] Static schema representation

- ○

  [A] Full fidelity schema representation, [B] Well-defined schema representation

- ○

  [A] SQLC schema representation, [B] Full fidelity schema representation

- ○

  [A] Well-defined schema representation, [B] Full fidelity schema representation

  (Correct)

Explanation
Azure Cosmos DB analytical store is a fully isolated column store for enabling large-scale analytics against operational data in your Azure Cosmos DB, without any impact to your transactional workloads.

There are two constraints that apply to the schema inferencing done by the autosync process as it transparently maintains the schema in the analytical store based on items added or updated in the transactional store:

- You can have a maximum of unique 1000 properties at any nesting level within the items stored in a transactional store. Any property above this and its associated values will not be present in the analytical store.

- Property names must be unique when compared in a case insensitive manner. For example, the properties `{"name": "Franklin Ye"}` and `{"Name": "Franklin Ye"}` cannot exit at the same nesting level in the same item or different items within a container given that “name” and “Name” are not unique when compared in a case insensitive manner.

There are two modes of schema representation for data stored in the analytical store. These modes have tradeoffs between the simplicity of a columnar representation, handling the polymorphic schemas, and simplicity of query experience:

- Well-defined schema representation

- Full fidelity schema representation

For SQL (Core) API accounts, when analytical store is enabled, the default schema representation in the analytical store is well-defined. Whereas for Azure Cosmos DB API for MongoDB accounts, the default schema representation in the analytical store is full fidelity schema representation. (If you have scenarios requiring a different schema representation than the default offering for each of these APIs, reach out to the Azure Cosmos DB team to enable it.)

Well-defined schema representation

The well-defined schema representation creates a simple tabular representation of the schema-agnostic data in the transactional store. The well-defined schema representation has the following considerations:

- The first document defines the base schema and property must always have the same type across all documents. The only exceptions are:

    - From null to any other data type.The first non-null occurrence defines the column data type. Any document not following the first non-null datatype won't be represented in analytical store.

    - From `float` to `integer`. All documents will be represented in analytical store.

- From `integer` to `float`. All documents will be represented in analytical store. However, to read this data with Azure Synapse SQL serverless pools, you must use a WITH clause to convert the column to `varchar`. And after this initial conversion, it is possible to convert it again to a number. Please check the example below, where **num** initial value was an integer and the second one was a float.

```
SQL
SELECT CAST (num as float) as num
FROM OPENROWSET(PROVIDER = 'CosmosDB',
                CONNECTION = '<your-connection',
                OBJECT = 'IntToFloat',
                SERVER_CREDENTIAL = 'your-credential'
)
WITH (num varchar(100)) AS [IntToFloat]
```

- Properties that don't follow the base schema data type won't be represented in analytical store. For example, consider the documents below: the first one defined the analytical store base schema. The second document, where `id` is `"2"`, **doesn't** have a well-defined schema since property `"code"` is a string and the first document has `"code"` as a number. In this case, the analytical store registers the data type of `"code"` as `integer` for lifetime of the container. The second document will still be included in analytical store, but its `"code"` property will not.

  - `{"id": "1", "code":123}`

  - `{"id": "2", "code": "123"}`

- Array types must contain a single repeated type. For example, `{"a": ["str",12]}` is not a well-defined schema because the array contains a mix of integer and string types.

- Expect different behaviour in regard to different types in well defined schema:

  - Spark pools in Azure Synapse will represent these values as `undefined`.

  - SQL serverless pools in Azure Synapse will represent these values as `NULL`.

- Expect different behaviour in regard to explicit `null` values:

  - Spark pools in Azure Synapse will read these values as `0` (zero). And it will change to `undefined` as soon as the column has a non-null value.

  - SQL serverless pools in Azure Synapse will read these values as `NULL`.

- Expect different behaviour in regard to missing columns:

- Spark pools in Azure Synapse will represent these columns as `undefined`.

- SQL serverless pools in Azure Synapse will represent these columns as `NULL`.

Full-fidelity schema representation

The full-fidelity schema representation creates a more complex tabular representation of the schema-agnostic data in the transactional store as it copies it to the analytical store. The full-fidelity schema representation has the top-level properties of the documents exposed as columns when queried from both Synapse SQL and Synapse Spark along with a JSON representation of the properties values contained within as column values. This is extended to include the data type of the properties along with their property values and as such can better handle polymorphic schemas of operational. With this schema representation, no items are dropped from the analytical store due to the need to meet the well-defined schema rules. For example, let's take the following sample document in the transactional store:

```
JSON
{
name: "John Doe",
age: 32,
profession: "Doctor",
address: {
  streetNo: 15850,
  streetName: "NE 40th St.",
  zip: 98052
},
salary: 1000000
}
```

The leaf property `streetNo` within the nested object `address` will be represented in the analytical store schema as a column `address.object.streetNo.int32`. The datatype is added as a suffix to the column. This way, if another document is added to the transactional store where the value of leaf property `streetNo` is "123" (note it's a string), the schema of the analytical store automatically evolves without altering the type of a previously written column. A new column added to the analytical store as `address.object.streetNo.string` where this value of "123" is stored.

https://docs.microsoft.com/en-us/azure/cosmos-db/analytical-store-introduction

Question 49: **Incorrect**

Scenario: You have been contracted by Wayne Enterprises, a company owned by Bruce Wayne with market value of over twenty seven million dollars. Bruce founded Wayne

Enterprises shortly after he created the Wayne Foundation and he became the president and chairman of the company.

Bruce has come to you because his IT team plans to use Microsoft Azure Synapse Analytics.

The IT team is lead by Oswald Cobblepot and his team created a table named SalesFact in an enterprise data warehouse in Azure Synapse Analytics. SalesFact contains sales data from the past 36 months and has the following characteristics:
• Is partitioned by month
• Contains one billion rows
• Has clustered columnstore indexes
At the beginning of each month, Bruce requires that the team removes data from SalesFact that is older than 36 months as quickly as possible.
The following is a list of items which Oswald believe he should action, but he is not sure which to use, nor which order to execute the actions in.

a. Switch the partition containing the stale data from SaleFact to SalesFact_Work.

b. Truncate the partition containing the stale data.

c. Drop the SalesFact_Work table.

d. Create an empty table named SalesFact_Work that has the same schema as SalesFact.

e. Execute s DELETE statement where the value in the Date column is more than 36 months ago.

f. Copy the data to a new table by using `CREATE TABLE AS SELECT`.

Which actions should Oswald perform in sequence in a stored procedure?

- ○

  f → a → b

- ◉

  f → b → c

  (Incorrect)

- ○

f → e

- ○

  d → a → c

  (Correct)

Explanation
The correct items and sequence is d → a → c.

Step 1: Create an empty table named SalesFact_work that has the same schema as SalesFact.
Step 2: Switch the partition containing the stale data from SalesFact to SalesFact_Work.
SQL Data Warehouse supports partition splitting, merging, and switching. To switch partitions between two tables, you must ensure that the partitions align on their respective boundaries and that the table definitions match.
Loading data into partitions with partition switching is a convenient way stage new data in a table that is not visible to users the switch in the new data.
Step 3: Drop the SalesFact_Work table.

What are table partitions?

Table partitions enable you to divide your data into smaller groups of data. In most cases, table partitions are created on a date column. Partitioning is supported on all dedicated SQL pool table types; including clustered columnstore, clustered index, and heap. Partitioning is also supported on all distribution types, including both hash or round robin distributed.

Partitioning can benefit data maintenance and query performance. Whether it benefits both or just one is dependent on how data is loaded and whether the same column can be used for both purposes, since partitioning can only be done on one column.

Benefits to loads

The primary benefit of partitioning in dedicated SQL pool is to improve the efficiency and performance of loading data by use of partition deletion, switching and merging. In most cases data is partitioned on a date column that is closely tied to the order in which the data is loaded into the SQL pool. One of the greatest benefits of using partitions to maintain data is the avoidance of transaction logging. While simply inserting, updating, or deleting data can be the most straightforward approach, with a little thought and effort, using partitioning during your load process can substantially improve performance.

Partition switching can be used to quickly remove or replace a section of a table. For example, a sales fact table might contain just data for the past 36 months. At the end of every month, the oldest month of sales data is deleted from the table. This data could be deleted by using a delete statement to delete the data for the oldest month.

However, deleting a large amount of data row-by-row with a delete statement can take too much time, as well as create the risk of large transactions that take a long time to rollback if something goes wrong. A more optimal approach is to drop the oldest partition of data. Where deleting the individual rows could take hours, deleting an entire partition could take seconds.

Benefits to queries

Partitioning can also be used to improve query performance. A query that applies a filter to partitioned data can limit the scan to only the qualifying partitions. This method of filtering can avoid a full table scan and only scan a smaller subset of data. With the introduction of clustered columnstore indexes, the predicate elimination performance benefits are less beneficial, but in some cases there can be a benefit to queries.

For example, if the sales fact table is partitioned into 36 months using the sales date field, then queries that filter on the sale date can skip searching in partitions that don't match the filter.

Sizing partitions

While partitioning can be used to improve performance some scenarios, creating a table with **too many** partitions can hurt performance under some circumstances. These concerns are especially true for clustered columnstore tables.

For partitioning to be helpful, it is important to understand when to use partitioning and the number of partitions to create. There is no hard fast rule as to how many partitions are too many, it depends on your data and how many partitions you loading simultaneously. A successful partitioning scheme usually has tens to hundreds of partitions, not thousands.

When creating partitions on **clustered columnstore** tables, it is important to consider how many rows belong to each partition. For optimal compression and performance of clustered columnstore tables, a minimum of 1 million rows per distribution and partition is needed. Before partitions are created, dedicated SQL pool already divides each table into 60 distributed databases.

Any partitioning added to a table is in addition to the distributions created behind the scenes. Using this example, if the sales fact table contained 36 monthly partitions, and given that a dedicated SQL pool has 60 distributions, then the sales fact table should contain 60 million rows per month, or 2.1 billion rows when all months are populated. If

a table contains fewer than the recommended minimum number of rows per partition, consider using fewer partitions in order to increase the number of rows per partition.
https://docs.microsoft.com/en-us/azure/sql-data-warehouse/sql-data-warehouse-tables-partition

Question 50: **Incorrect**
Identify the missing word(s) in the following sentence within the context of Microsoft Azure.

Setting Global parameters in an Azure Data Factory pipeline … [?]

- ●

  references to an attribute like a dataset or data flow, this reroutes default parameter values through the resource parameter.

  (Incorrect)

- ○

  cannot be overridden if you wish to use continuous integration and deployment process.

- ○

  allows you to use constants for consumption in pipeline expressions.

  (Correct)

- ○

  None of the listed options.

Explanation
Global parameters in Azure Data Factory

Setting Global parameters in an Azure Data Factory pipeline, allows you to use constants for consumption in pipeline expressions. A use-case for setting global parameters is when you have multiple pipelines where the parameters names and values are identical. If you use the continuous integration and deployment process with Azure Data Factory, the global parameters can be overridden if you wish so, for each and every environment that you have created.

Using global parameters in a pipeline

When using global parameters in a pipeline in Azure Data Factory, it is mostly referenced in pipeline expressions. For example, if a pipeline references to a resource like a dataset or data flow, you can pass down the global parameter value through the resource parameter. The command or reference of global parameters in Azure Data Factory flows as follows: `pipeline().globalParameters`.

Global parameters in CI/CD

When you integrate global parameters in a pipeline using CI/CD with Azure Data Factory, you have two ways in order to do so:

- Include global parameters in the Azure Resource Manager template

- Deploy global parameters via a PowerShell script

In most CI/CD practices, it is beneficial to include global parameters in the Azure Resource Manager template. The reason why it's recommended is of the native integration with CI/CD where global parameters are added as an Azure Resource Manager Template parameter due to changes in several environments that are worked in. In order to enable global parameters in an Azure Resource Manager template, you navigate to the management hub. You do have to be aware that once you add global parameters to an Azure Resource Manager template, it adds an Azure Data Factory level setting, which can override other settings like git configs.

The use case for deploying global parameters through a PowerShell script, could be because you might have the above mentioned settings enabled in an elevated environment like UAT or PROD.

Parameterize mapping dataflows

Within Azure Data Factory, you are able to use mapping data flows and therefore, enabling you to use parameters. If you set parameters inside a data flow definition, you can use the parameters in expressions. The parameter values will be set by the calling pipeline through the Execute Data Flow activity.

There are three options for setting the values in the data flow activity expressions:

- Use the pipeline control flow expression language to set a dynamic value

- Use the data flow expression language to set a dynamic value

- Use either expression language to set a static literal value

The reason for parameterizing mapping data flows, is to make sure that your data flows are generalized, flexible, and reusable.

https://docs.microsoft.com/en-us/azure/data-factory/parameterize-linked-services

Question 51: **Incorrect**

Identify the missing word(s) in the following sentence within the context of Microsoft Azure.

Linux foundation [?] is an open-source storage layer for Spark that enables relational database capabilities for batch and streaming data. By using [?], you can implement a *data lakehouse* architecture in Spark to support SQL_based data manipulation semantics with support for transactions and schema enforcement. The result is an analytical data store that offers many of the advantages of a relational database system with the flexibility of data file storage.

- ●

  River Delta

  (Incorrect)

- ○

  Delta Lake

  (Correct)

- ○

  Data Lake

- ○

  Data Stream

- ○

  Delta Ocean

- ○

  Data Ocean

Explanation

Linux foundation Delta Lake is an open-source storage layer for Spark that enables relational database capabilities for batch and streaming data. By using Delta Lake, you can implement a data lakehouse architecture in Spark to support SQL_based data *manipulation semantics with support for transactions and schema enforcement. The result is an analytical data store that offers many of the advantages of a relational database system with the flexibility of data file storage in a data lake.*

https://delta.io/

Get Started with Delta Lake

Delta Lake is an open-source storage layer that adds relational database semantics to Spark-based data lake processing. Delta Lake is supported in Azure Synapse Analytics Spark pools for PySpark, Scala, and .NET code.

The benefits of using Delta Lake in Azure Databricks include:

• **Relational tables that support querying and data modification**. With Delta Lake, you can store data in tables that support *CRUD* (create, read, update, and delete) operations. In other words, you can *select*, *insert*, *update*, and *delete* rows of data in the same way you would in a relational database system.

• **Support for *ACID* transactions**. Relational databases are designed to support transactional data modifications that provide *atomicity* (transactions complete as a single unit of work), *consistency* (transactions leave the database in a consistent state), *isolation* (in-process transactions can't interfere with one another), and *durability* (when a transaction completes, the changes it made are persisted). Delta Lake brings this same transactional support to Spark by implementing a transaction log and enforcing serializable isolation for concurrent operations.

• **Data versioning and *time travel***. Because all transactions are logged in the transaction log, you can track multiple versions of each table row, and even use the *time travel* feature to retrieve a previous version of a row in a query.

• **Support for batch and streaming data**. While most relational databases include tables that store static data, Spark includes native support for streaming data through the Spark Structured Streaming API. Delta Lake tables can be used as both *sinks* (destinations) and *sources* for streaming data.

• **Standard formats and interoperability**. The underlying data for Delta Lake tables is stored in Parquet format, which is commonly used in data lake ingestion pipelines.

https://learn.microsoft.com/en-us/azure/databricks/delta/

Question 52: **Incorrect**
Azure Data Factory is composed of four core components. These components work together to provide the platform on which you can compose data-driven workflows with steps to move and transform data.

Which component is best described by:

"It has information on the different data sources and Data Factory uses this information to connect to data originating sources. It is mainly used to locate the data stores in the machines and also represent the compute services for the activity to be executed, such as running spark jobs on spark clusters or running hive queries using hive services from the cloud."

- ◉

  Pipeline

  (Incorrect)

- ○

  Activity

- ○

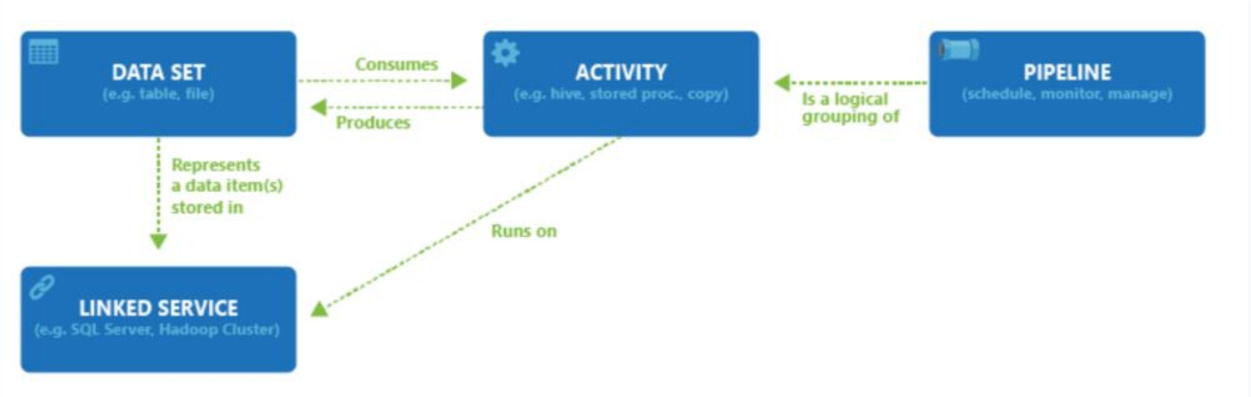
  Dataset

- ○

  Linked service

  (Correct)

Explanation
An Azure subscription might have one or more Azure Data Factory instances. Azure Data Factory is composed of four core components. These components work together to provide the platform on which you can compose data-driven workflows with steps to move and transform data.


DATA SET
(e.g. table, file)
Consumes
Produces
ACTIVITY
(e.g. hive, stored proc., copy)
Is a logical grouping of
PIPELINE
(schedule, monitor, manage)
Represents a data item(s) stored in
Runs on
LINKED SERVICE
(e.g. SQL Server, Hadoop Cluster)

• **Pipeline:** It is created to perform a specific task by composing the different activities in the task in a single workflow. Activities in the pipeline can be data ingestion (Copy data to Azure) -> data processing (Perform Hive Query). Using pipeline as a single task user can schedule the task and manage all the activities in a single process also it is used to run the multiple operation parallel. Multiple activities can be logically grouped together with an object referred to as a **Pipeline**, and these can be *scheduled* to execute, or a *trigger* can be defined that determines when a pipeline execution needs to be kicked off. There are different types of triggers for different types of events.

• **Activity:** It is a specific action performed on the data in a pipeline like the transformation or ingestion of the data. Each pipeline can have one or more activities in it. If the data is copied from one source to destination using Copy Monitor then it is a data movement activity. If data transformation is performed on the data using a hive query or spark job then it is a data transformation activity.

• **Datasets:** It is basically collected data users required which are used as input for the ETL process. Datasets have different formats; they can be in JSON, CSV, ORC, or text format.

• **Linked services:** It has information on the different data sources and the data factory uses this information to connect to data originating sources. It is mainly used to locate the data stores in the machines and also represent the compute services for the activity to be executed like running spark jobs on spark clusters or running hive queries using the hive services from the cloud.

https://www.educba.com/azure-data-factory/

Question 53: **Correct**

Integration Runtime (IR) is the compute infrastructure used by Azure Data Factory. It provides data integration capabilities across different network environments. Data Factory offers three types of Integration Runtime.

These three IR types are:

• Azure

• Self-hosted

• Azure-SSIS

Which does not have Private network support?

- ○

  Azure

- ●

  None of the listed options.

  (Correct)

- ○

  All the listed options.

- ○

  Azure-SSIS

- ○

  Self-hosted

Explanation

None of the listed options do not support private networks - this is a tricky question which uses double-negatives to create the positive - which means all the listed options support private networks. Sometimes you will run into questions that are intended to trick you. More of a question of grammar than actual Azure knowledge.

The question is asking which of the listed DO NOT support Private networks.

They all do, therefore 'None of the listed options DO NOT support Private networks', which is the same as saying 'All the options DO support Private networks'. This question is intended to trip you up so you read what is being asked.

You may come across questions in your exam intended to trick you and you must read the question thoroughly.

In Data Factory, an activity defines the action to be performed. A linked service defines a target data store or a compute service. An integration runtime provides the infrastructure for the activity and linked services.

Integration Runtime is referenced by the linked service or activity, and provides the compute environment where the activity either runs on or gets dispatched from. This way, the activity can be performed in the region closest possible to the target data store or compute service in the most performant way while meeting security and compliance needs.

In short, the Integration Runtime (IR) is the compute infrastructure used by Azure Data Factory. It provides the following data integration capabilities across different network environments, including:

- **Data Flow**: Execute a Data Flow in managed Azure compute environment.

- **Data movement**: Copy data across data stores in public network and data stores in private network (on-premises or virtual private network). It provides support for built-in connectors, format conversion, column mapping, and performant and scalable data transfer.

- **Activity dispatch**: Dispatch and monitor transformation activities running on a variety of compute services such as Azure Databricks, Azure HDInsight, Azure Machine Learning, Azure SQL Database, SQL Server, and more.

- **SSIS package execution**: Natively execute SQL Server Integration Services (SSIS) packages in a managed Azure compute environment.

Whenever an Azure Data Factory instance is created, a default Integration Runtime environment is created that supports operations on cloud data stores and compute services in public network. This can be viewed when the integration runtime is set to Auto-Resolve.

Integration runtime types

Data Factory offers three types of Integration Runtime, and you should choose the type that best serve the data integration capabilities and network environment needs you are looking for. These three types are:

- Azure

- Self-hosted

• Azure-SSIS

You can explicitly define the Integration Runtime setting in the **connectVia** property, if this is not defined, then the default Integration Runtime is used with the property set to Auto-Resolve.

The following describes the capabilities and network support for each of the integration runtime types:

IR type: Azure

Public network: Data Flow, Data movement and Activity dispatch

Private network: Data Flow, Data movement and Activity dispatch

IR type: Self-hosted

Public network: Data movement Activity dispatch

Private network: Data movement Activity dispatch

IR type: Azure-SSIS

Public network: SSIS package execution

Private network: SSIS package execution

https://docs.microsoft.com/en-us/azure/data-factory/concepts-integration-runtime

Question 54: **Incorrect**
Scenario: The company you work at is in the Healthcare industry, which in turn is working with a specific health care provider. This healthcare provider only wants doctors and nurses to be able to access medical records. The billing department should not have access to view this data.

Which type of security would typically be best used in for this scenario?

- ◉

  Dynamic Data Masking

(Incorrect)

- ○

  Table-level security

- ○

  Column-level security

  (Correct)

- ○

  Row-level security

Explanation

Authentication is the process of validating credentials as you access resources in a digital infrastructure. This ensures that you can validate that an individual, or a service that wants to access a service in your environment can prove who they are. Azure Synapse Analytics provides several different methods for authentication.

An important thing to note in the official exam questions is that there may be more that one available response that would work, but they often ask which is the best. While this may seem subjective, it is a real scenario you will likely encounter in the exam.

*While both data masking and column-level security would accomplish the goal of protecting the sensitive data, **Column-level security** simplifies the design and coding of **security** in your application, allowing you to restrict **column** access to protect sensitive **data**, not just hide the sensitive data.*

Column level security in Azure Synapse Analytics

Generally speaking, column level security is simplifying a design and coding for the security in your application. It allows you to restrict column access in order to protect sensitive data. For example, if you want to ensure that a specific user 'Leo' can only access certain columns of a table because he's in a specific department. The logic for 'Leo' only to access the columns specified for the department he works in, is a logic that is located in the database tier, rather on the application level data tier. If he needs to access data from any tier, the database should apply the access restriction every time he tries to access data from another tier. The reason for doing so, is to make sure that your security is reliable and robust since we're reducing the surface area of the overall security system. Column level security will also eliminate the necessity for the introduction of view, where you would filter out columns, to impose access restrictions on 'Leo'

The way to implement column level security, is by using the `GRANT` T-SQL statement. Using this statement, SQL and Azure Active Directory (AAD) support the authentication.

The syntax to use for implementing column level security looks as follows:

```
SQL
GRANT <permission> [ ,...n ] ON
[ OBJECT :: ][ schema_name ]. object_name [ ( column [ ,...n ] ) ] // specifying the column access
TO <database_principal> [ ,...n ]
[ WITH GRANT OPTION ]
[ AS <database_principal> ]
<permission> ::=
SELECT
| UPDATE
<database_principal> ::=
Database_user // specifying the database user
| Database_role // specifying the database role
| Database_user_mapped_to_Windows_User
| Database_user_mapped_to_Windows_Group
```

So when would you use column-level security? Let's say that you are a financial services firm, and can only have account manager allowed to have access to a customer's social security number, phone numbers or other personal identifiable information. It is imperative to distinguish the role of an account manager versus the manager of the account managers.

Another use case might be related to the Healthcare Industry. Let's say you have a specific health care provider. This healthcare provider only wants doctors and nurses to be able to access medical records. The billing department should not have access to view this data. Column-level security would typically be the option to use.

Row level security in Azure Synapse Analytics

Row-level security (RLS) can help you to create a group membership or execution context in order to control not just columns in a database table, but actually, the rows. RLS, just like column-level security, can simply help and enable your design and coding of your application security. However, compared to column-level security where it's focused on the columns (parameters), RLS helps you implement restrictions on data row access. Let's say that your employee can only access rows of data that are important of the department, you should implement RLS. If you want to restrict for example, customer's data access that is only relevant to the company, you can implement RLS. The restriction on access of the rows, is a logic that is located in the database tier, rather on the application level data tier. If 'Leo' needs to access data from any tier, the database should apply the access restriction every time he tries to access data from another tier. The reason for doing so, is to make sure that your security is reliable and robust since we're reducing the surface area of the overall security system.

The way to implement RLS is by using the `CREATE SECURITY POLICY[!INCLUDEtsql]` statement. The predicates are created as inline table-valued functions. It is imperative to understand that within Azure Synapse, it only supports filter predicates. If you need to use a block predicate, you won't be able to find support at this moment within in Azure synapse.

Description of row level security in relation to filter predicates

RLS within Azure Synapse supports one type of security predicates, which are Filter predicates, not block predicates.
What filter predicates do, are silently filtering the rows that are available for read operations such as `SELECT`, `UPDATE`, `DELETE`.

The access to row-level data in a table, is restricted as an inline table-valued function, which is a security predicate. This table-valued function will then be invoked and enforced by the security policy that you need. An application, is not aware of rows that are filtered from the result set for filter predicates. So what will happen is that if all rows are filtered, a null set is returned.

When you are using filter predicates, it will be applied when data is read from the base table. The filter predicate affects all get operations such as `SELECT`, `DELETE`, `UPDATE`. You are unable to select or delete rows that have been filtered. It is not possible for you to update a row that has been filtered. What you can do, is update rows in a way that they will be filtered afterwards.

Permissions

If you want to create, alter or drop the security policies, you would have to use the `ALTER ANY SECURITY POLICY` permission. The reason for that is when you are creating or dropping a security policy it requires `ALTER` permissions on the schema.

In addition to that, there are other permissions required for each predicate that you would add:

- `SELECT` and `REFERENCES` permissions on the inline table-valued function being used as a predicate.

- `REFERENCES` permission on the table that you target to be bound to the policy.

- `REFERENCES` permission on every column from the target table used as arguments.

Once you've set up the security policies, they will apply to all the users (including dbo users in the database) Even though DBO users can alter or drop security policies, their changes to the security policies can be audited. If you have special circumstances where highly privileged users, like a sysadmin or `db_owner`, need to see all rows to troubleshoot or validate data, you would still have to write the security policy in order to allow that.

If you have created a security policy where `SCHEMABINDING = OFF`, in order to query the target table, the user must have the `SELECT` or `EXECUTE` permission on the predicate function. They also need permissions to any additional tables, views, or functions used within the predicate function. If a security policy is created with `SCHEMABINDING = ON` (the default), then these permission checks are bypassed when users query the target table.

Best practices

There are some best practices to take in mind when you want to implement RLS. We recommended creating a separate schema for the RLS objects. RLS objects in this context would be the predicate functions, and security policies. Why is that a best practice? It helps to separate the permissions that are required on these special objects from the target tables. In addition to that, separation for different policies and predicate functions may be needed in multi-tenant-databases. However, it is not a standard for every case.

Another best practice to bear in mind is that the `ALTER ANY SECURITY POLICY` permission should only be intended for highly privileged users (such as a security policy manager). The security policy manager should not require **`SELECT`** permission on the tables they protect.

In order to avoid potential runtime errors, you should take in mind type conversions in predicate functions that you write. Also, you should try to avoid recursion in predicate functions. The reason for this is to avoid performance degradation. Even though the query optimizer will try to detect the direct recursions, there is no guarantee to find the indirect recursions. With an indirect recursion we mean where a second function call the predicate function.

It would also be recommended to avoid the use of excessive table joins in predicate functions. This would maximize performance.

Generally speaking when it comes to the logic of predicates, you should try to avoid logic that depends on session-specific SET options. Even though this is highly unlikely to be used in practical applications, predicate functions whose logic depends on certain session-specific **SET** options can leak information if users are able to execute arbitrary queries. For example, a predicate function that implicitly converts a string to **datetime** could filter different rows based on the `SET DATEFORMAT` option for the current session.

https://docs.microsoft.com/en-us/azure/synapse-analytics/sql-data-warehouse/column-level-security

Question 55: **Incorrect**

Across all organizations and industries, common use cases for Azure Synapse Analytics are which of the following? (Select all that apply)

- ☑

Data exploration and discovery

(Correct)

- □

  Real time analytics

  (Correct)

- □

  Data integration

  (Correct)

- □

  AI learning troubleshooting

- □

  Modern data warehousing

  (Correct)

- □

  Integrated analytics

  (Correct)

- □

  Advanced analytics

  (Correct)

- □

  IoT device deployment

Explanation

Across all organizations and industries, the common use cases for Azure Synapse Analytics are identified by the need for:

Modern data warehousing

This involves the ability to integrate all data, including big data, to reason over data for analytics and reporting purposes from a descriptive analytics perspective, independent of its location or structure.

Advanced analytics

Enables organizations to perform predictive analytics using both the native features of Azure Synapse Analytics, and integrating with other technologies such as Azure Databricks.

Data exploration and discovery

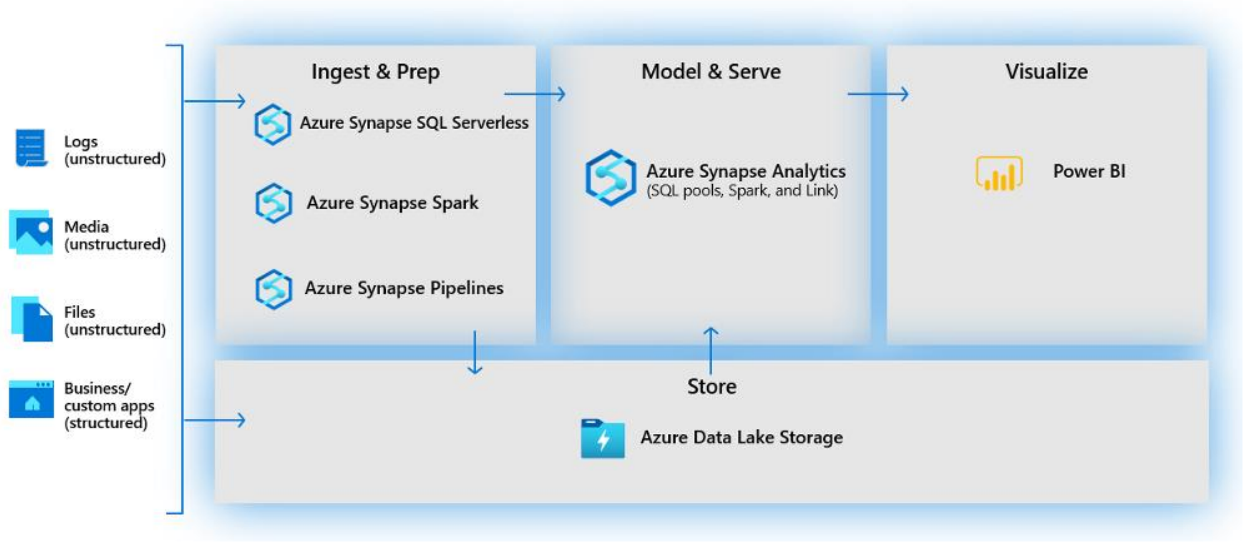
The SQL serverless functionality provided by Azure Synapse Analytics enables Data Analysts, Data Engineers and Data Scientist alike to explore the data within your data estate. This capability supports data discovery, diagnostic analytics, and exploratory data analysis.

Real time analytics

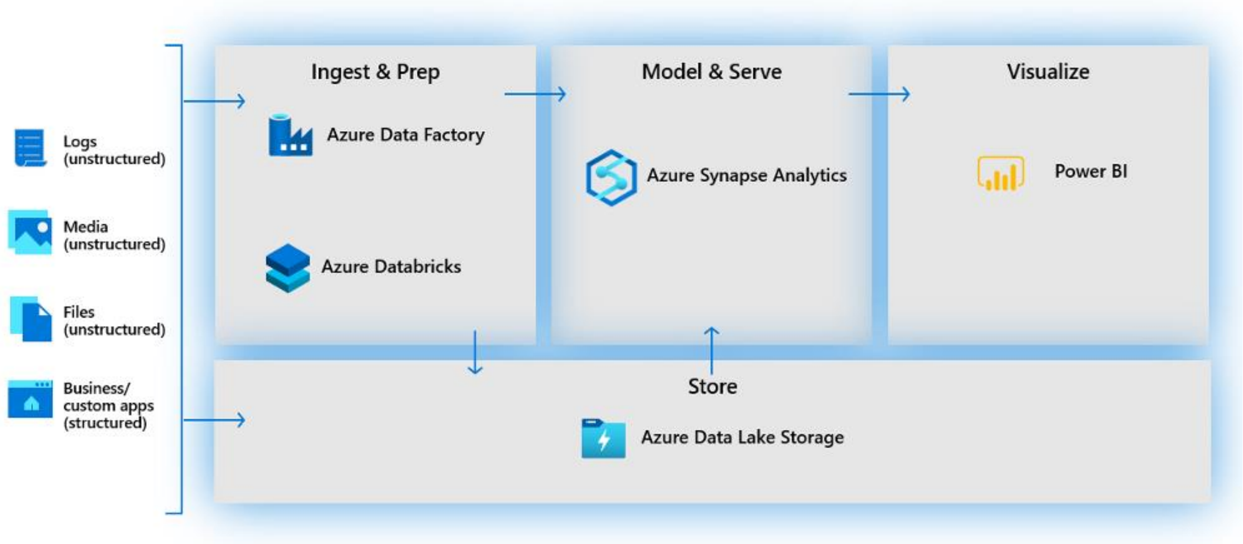
Azure Synapse Analytics can capture, store and analyze data in real-time or near-real time with features such as Azure Synapse Link, or through the integration of services such as Azure Stream Analytics and Azure Data Explorer.

Data integration

Azure Synapse Pipelines enables you to ingest, prepare, model and serve the data to be used by downstream systems. This can be used by components of Azure Synapse Analytics exclusively.



It can also interact with existing Azure services that you may already have in place for your existing analytical solutions.



Integrated analytics

With the variety of analytics that can be performed on the data at your disposal, putting together the services in a cohesive solution can be a complex operation. Azure Synapse Analytics removes this complexity by integrating the analytics landscape into on service. That way you can spend more time working with the data to bring business benefit, than spending much of your time provisioning and maintaining multiple systems to achieve the same outcomes.

https://docs.microsoft.com/en-us/azure/synapse-analytics/overview-what-is