Question 1: Skipped

Synapse Studio comes with an integrated notebook experience. The notebooks in Synapse studio, are a web interface that enables you to create, edit, or transform data in the files. It is based on a live code experience, including visualizations and narrative text.

True or False: You can access data in the primary storage account directly. There's no need to provide the secret keys.

- True
  (Correct)
- False

Explanation

Synapse Studio comes with an integrated notebook experience. The notebooks in Synapse studio, are a web interface that enables you to create, edit, or transform data in the files. It is based on a live code experience, including visualizations and narrative text.

If you'd like to experiment with your data and gain some insights about the data, notebooks are a good way to start and validate some of the ideas you might have.

The look and feel of the integrated notebook experience is similar to, for example, the jupyter notebooks in Azure Machine Learning Service or other IDEs you might use and interact with on your data.

If you navigate to the Synapse studio environment, you can find the notebooks in the Development Hub of the studio experience. To access the studio environment, you can navigate to the Azure Synapse Analytics Workspace and launch the studio. You'll also find that there are some notebook examples available through the Knowledge Centre.

The notebooks allow you to write multiple languages in one notebook by using the magic commands expressed by %%

The visual aspects of the notebooks are

• Support for Language Syntax highlight

• Syntax error

• Syntax code completion

• Export results

Within the notebook environment of the Azure Synapse Analytics Studio, you have the possibility tc create temporary tables across the multiple languages you might use.

In order to ingest data through notebooks you can use a linked service from the workspace, to, for example, an Azure Data Lake storage where then the keys and access are automatically passed through to the storage account where you have stored the file that you want to ingest or read out into a spark DataFrame.

You can access data in the primary storage account directly. There's no need to provide the secret keys. In the Data tab on the left hand-side in the Synapse Workspace, right-click on a file and select New notebook to see a new notebook with data extractor autogenerated.

With an Azure Synapse Studio notebook, you can:

• Get easily started.

• Keep data secure with built-in enterprise security features.

• Analyze data across raw formats (CSV, txt, JSON, etc.), processed file formats (parquet, Delta Lake, ORC, etc.), and SQL tabular data files against Spark and SQL.

• Be productive with enhanced authoring capabilities and built-in data visualization.

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

---

Question 2: Skipped

How can parameters be passed into an Azure Databricks notebook from Azure Data Factory?

- Use the new API endpoint option on a notebook in Databricks and provide the parameter name.
- Deploy the notebook as a web service in Databricks, defining parameter names and types.
- Use notebook widgets to define parameters that can be passed into the notebook. **(Correct)**
- Render the notebook to an API endpoint in Databricks, defining parameter names and types.

Explanation

You can configure parameters by using widgets on the Databricks notebook. You then pass in parameters with those names via a Databricks notebook activity in Data Factory.

https://docs.databricks.com/notebooks/widgets.html

Question 3: Skipped

Azure Data Factory provides a variety of methods for ingesting data, and also provides a range of methods to perform transformations.

These methods are:

• Mapping Data Flows

• Compute Resources

• SSIS Packages

Mapping Data Flows provides a number of different transformations types that enable you to modify data. They are broken down into the following categories:

• Schema modifier transformations

• Row modifier transformations

• Multiple inputs/outputs transformations

Which of the following are valid transformations available in the Mapping Data Flow? (Select all that apply)

- ☐ Aggregate **(Correct)**
- ☐ Round
- ☐ Filter **(Correct)**
- ☐ Exists **(Correct)**
- ☐ Trim
- ☐ Conditional split **(Correct)**
- ☐

- Union
  (Correct)
- Derived column
  (Correct)
- Merge
- Flatten
  (Correct)
- Join
  (Correct)
- Between
- Lookup
  (Correct)
- Alter row
  (Correct)
- Avg

Explanation

Just as Azure Data Factory provides a variety of methods for ingesting data, it also provides a range of methods to perform transformations. You can pick a method that matches the skillsets of your team or takes advantage of existing technologies that you already have in your data estate. There is also the opportunity to perform transformations without writing code at all using the Mapping Data Flow.

Transforming data using Mapping Data Flow

Mapping Data Flows provide an environment for building a wide range of data transformations visually without the need to use code. The resulting data flows that are created are subsequently executed on scaled-out Apache Spark clusters that are automatically provisioned when you execute the Mapping Data Flow. Mapping Data Flows also provides the capability to monitor the execution of the transformations so that you can view how the transformations are progressing, or to understand any errors that may occur

Mapping Data Flows provides a number of different transformations types that enable you to modify data. They are broken down into the following categories:

• Schema modifier transformations

• Row modifier transformations

• Multiple inputs/outputs transformations

Below is a list of transformations that is available in the Mapping Data Flows:

Name & Category: Aggregate - Schema modifier

Description: Define different types of aggregations such as SUM, MIN, MAX, and COUNT grouped by existing or computed columns.

Name & Category: Alter row - Row modifier

Description: Set insert, delete, update, and upsert policies on rows. You can add one-to-many conditions as expressions. These conditions should be specified in order of priority, as each row will be marked with the policy corresponding to the first-matching expression. Each of those conditions can result in a row (or rows) being inserted, updated, deleted, or upserted. Alter Row can produce both DDL & DML actions against your database.

Name & Category: Conditional split - Multiple inputs/outputs

Description: Route rows of data to different streams based on matching conditions.

Name & Category: Derived column - Schema modifier

Description: Generate new columns or modify existing fields using the data flow expression language.

Name & Category: Exists - Multiple inputs/outputs

Description: Check whether your data exists in another source or stream.

Name & Category: Filter - Row modifier

Description: Filter a row based upon a condition.

Name & Category: Flatten - Schema modifier

Description: Take array values inside hierarchical structures such as JSON and unroll them into individual rows.

Name & Category: Join - Multiple inputs/outputs

Description: Combine data from two sources or streams.

Name & Category: Lookup - Multiple inputs/outputs

Description: Enables you to reference data from another source.

Name & Category: New branch - Multiple inputs/outputs

Description: Apply multiple sets of operations and transformations against the same data stream.

Name & Category: Pivot - Schema modifier

Description: An aggregation where one or more grouping columns has distinct row values transformed into individual columns.

Name & Category: Select - Schema modifier

Description: Alias columns and stream names, and drop or reorder columns.

Name & Category: Sink – N/A

Description: A final destination for your data.

Name & Category: Sort - Row modifier

Description: Sort incoming rows on the current data stream.

Name & Category: Source – N/A

Description: A data source for the data flow.

Name & Category: Surrogate key - Schema modifier

Description: Add an incrementing non-business arbitrary key value.

Name & Category: Union - Multiple inputs/outputs

Description: Combine multiple data streams vertically.

Name & Category: Unpivot - Schema modifier

Description: Pivot columns into row values.

Name & Category: Window - Schema modifier

Description: Define window-based aggregations of columns in your data streams.

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

---

Question 4: Skipped

Scenario: The company you work at stores several website asset types in Azure Storage. These types include images and videos. Which of the following is the best way to secure browser apps to lock `GET` requests?

- Lock `GET` requests down to specific domains using `CORS`.
  (Correct)
- Use Private Endpoints between the VMs and the company websites.
- Use Private Link on the company's websites.
- Lock `GET` requests down to specific domains using Vault.

Explanation
CORS support

Many companies store several website asset types in Azure Storage. These types include images and videos. To secure browser apps, it is recommended to lock `GET` requests down to specific domains.

Azure Storage supports cross-domain access through cross-origin resource sharing (CORS). CORS uses `HTTP` headers so that a web application at one domain can access resources from a server at a different domain. By using CORS, web apps ensure that they load only authorized content from authorized sources.

CORS support is an optional flag you can enable on Storage accounts. The flag adds the appropriate headers when you use `HTTP GET` requests to retrieve resources from the Storage account.

https://docs.microsoft.com/en-us/rest/api/storageservices/cross-origin-resource-sharing--cors--support-for-the-azure-storage-services

---

Question 5: Skipped

Activities within Azure Data Factory define the actions that will be performed on the data and there are three categories including:

• Data movement activities

• Data transformation activities

• Control activities

Pipelines in Data Factory are defined in JSON format as follows:

```
JSON
{
"name": "PipelineName",
"properties":
{
"description": "pipeline description",
"activities":
[
],
"parameters": {
}
}
}
```

Which of the JSON properties are required? (Select all that apply)

- ☐ parameters
- ☐ activities **(Correct)**
- ☐ name **(Correct)**
- ☐ description

Explanation

Activities within Azure Data Factory define the actions that will be performed on the data and there are three categories including:

• Data movement activities

• Data transformation activities

• Control activities

Activities and pipelines

Defining pipelines

Here is how a pipeline is defined in JSON format:

```
JSON
{
"name": "PipelineName",
"properties":
{
"description": "pipeline description",
"activities":
[
],
"parameters": {
}
}
}
```

The following describes properties in the above JSON:

Property: name

Name of the activity.

Required: Yes

Property: description

Text describing what the pipeline is used for.

Required: No

Property: activities

The activities section can have one or more activities defined within it..

Required: Yes

Property: parameters

The parameters section can have one or more parameters defined within the pipeline, making your pipeline flexible for reuse.

Required: No

https://docs.microsoft.com/en-us/azure/data-factory/concepts-pipelines-activities

---

Question 6: Skipped

Init Scripts provide a way to configure cluster's nodes. It is recommended to favour Cluster Scoped Init Scripts over Global and Named scripts.

Which of the following is best described by:

"You specify the script in cluster's configuration by either writing it directly in the cluster configuration UI or storing it on DBFS and specifying the path in Cluster Create API. Any location under `DBFS /databricks` *folder except* `/databricks/init` *can be used for this purpose."*

- Interactive
- Global
- Cluster Scoped
  (Correct)
- Cluster Named

Explanation
Favour cluster scoped init scripts over global and named scripts

Init Scripts provide a way to configure cluster's nodes and to perform custom installs. Init scripts can be used in the following modes:

• **Global:** by placing the Init script in `/databricks/init` folder, you force the script's execution every time any cluster is created or restarted by users of the workspace.

• **Cluster Named (deprecated):** you can limit the init script to run only on for a specific cluster's creation and restarts by placing it in `/databricks/init/<cluster_name>` folder.

• **Cluster Scoped**: in this mode, the Init script is not tied to any cluster by its name and its automatic execution is not a virtue of its dbfs location. Rather, you specify the script in cluster's configuration by either writing it directly in the cluster configuration UI or storing it on Databricks File System (DBFS) and specifying the path in Cluster Create API. Any location under `DBFS /databricks` folder except `/databricks/init` can be used for this purpose, such as: `/databricks/<my-directory>/set-env-var.sh`

You should treat Init scripts with *extreme* caution because they can easily lead to intractable cluster launch failures. If you really need them, please use the **Cluster Scoped execution mode** as much as possible because:

• ADB executes the script's body in each cluster node. Thus, a successful cluster launch and subsequent operation are predicated on all nodal Init scripts executing in a timely manner without any errors and reporting a zero exit code. This process is highly error prone, especially for scripts downloading artifacts from an external service over unreliable and/or misconfigured networks.

• Because Global and Cluster Named Init scripts execute automatically due to their placement in a special DBFS location, it is easy to overlook that they could be causing a cluster to not launch. By specifying the Init script in the Configuration, there's a higher chance that you'll consider them while debugging launch failures.

Use cluster log delivery feature to manage logs

By default, Cluster logs are sent to default DBFS but you should consider sending the logs to a blob store location under your control using the Cluster Log Delivery feature. The Cluster Logs contain logs emitted by user code, as well as Spark framework's Driver and Executor logs. Sending them to a blob store controlled by yourself is recommended over default DBFS location because:

• ADB's automatic 30-day default DBFS log purging policy might be too short for certain compliance scenarios. A blob store location in your subscription will be free from such policies.

• You can ship logs to other tools only if they are present in your storage account and a resource group governed by you. The root DBFS, although present in your subscription, is launched inside a Microsoft Azure managed resource group and is protected by a read lock. Because of this lock, the logs are only accessible by privileged Azure Databricks framework code. However, constructing a pipeline to ship the logs to downstream log analytics tools requires logs to be in a lock-free location first.

https://github.com/Azure/AzureDatabricksBestPractices/blob/master/toc.md

---

Question 7: Skipped

Which workload management feature influences the order in which a request gets access to resources?

- Workload importance
  (Correct)
- Workload priority
- Workload isolation
- Workload classification

Explanation

Workload importance indexes write multiple data types and values per row of data and so compression functions are less likely to be able to reduce the size through pattern matching and offsets.

Workload importance

Workload importance influences the order in which a request gets access to resources. On a busy system, a request with higher importance has first access to resources. Importance can also ensure ordered access to locks. There are five levels of importance: low, below_normal, normal, above_normal, and high. Requests that don't set importance are assigned the default level of normal. Requests that have the same importance level have the same scheduling behaviour that exists today.

https://docs.microsoft.com/en-us/azure/synapse-analytics/sql-data-warehouse/sql-data-warehouse-workload-management

---

Question 8: Skipped

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.

Which are valid authentication methods in Azure Synapse Analytics? (Select all that apply)

- ☐ SAML
- ☐ Azure Key Vault
  (Correct)
- ☐ OAuth
- ☐ SQL Authentication
  (Correct)
- ☐ Azure Active Directory
  (Correct)
- ☐ SAS
  (Correct)
- ☐ SSL
- ☐ MFA
  (Correct)
- ☐ Managed identity
  (Correct)

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.

What needs to be authenticated

There are a variety of scenarios that means that authentication must take place to protect the data that is stored in your Azure Synapse Analytics estate.

The common form of authentication is that of individuals who want to access the data in the service. This is typically seen as an individual providing a username and password to authenticate against a service. However, this is also becoming more sophisticated with authentication requests working in combination with conditional access policies to further secure the authentication process with additional security steps.

What is less obvious is the fact that services must authenticate with other services so that they can operate seamlessly. An example of this is using an Azure Synapse Spark or serverless SQL pool to access data in an Azure Data Lake store. An authentication mechanism must take place in the background to ensure that Azure Synapse Analytics can access the data in the data lake in an authenticated manner.

Finally, there are situations where users and services operate together at the same time. Here you have a combination of both user and service authentication taking place under the hood to ensure that the user is getting access to the data seamlessly. An example of this is using Power BI to view reports in a dashboard that is being serviced by a dedicated SQL pool. Here you have multiple levels of authentication taking place that needs to be managed.

Types of security

The following are the types of authentication that you should be aware of when working with Azure Synapse Analytics.

Azure Active Directory

Azure Active Directory is a directory service that allows you to centrally maintain objects that can be secured. The objects can include user accounts and computer accounts. An employee of an organization will typically have a user account that represents them in the organizations Azure Active Directory tenant, and they then use the user account with a password to authenticate against other resources that are stored within the directory using a process known as single sign-on.

The power of Azure Active Directory is that they only have to login once, and Azure Active Directory will manage access to other resources based on the information held within it using pass through authentication. If a user and an instance of Azure Synapse Analytics are part of the same Azure Active Directory, it is possible for the user to access Azure Synapse Analytics without an apparent login. If managed correctly, this process is seamless as the administrator would have given the user authorization to access Azure Synapse Analytics dedicated SQL pool as an example.

In this situation, it is normal for an Azure Administrator to create the user accounts and assign them to the appropriate roles and groups in Azure Active Directory. The Data Engineer will then add the user, or a group to which the user belongs to access a dedicated SQL pool.

Managed identities

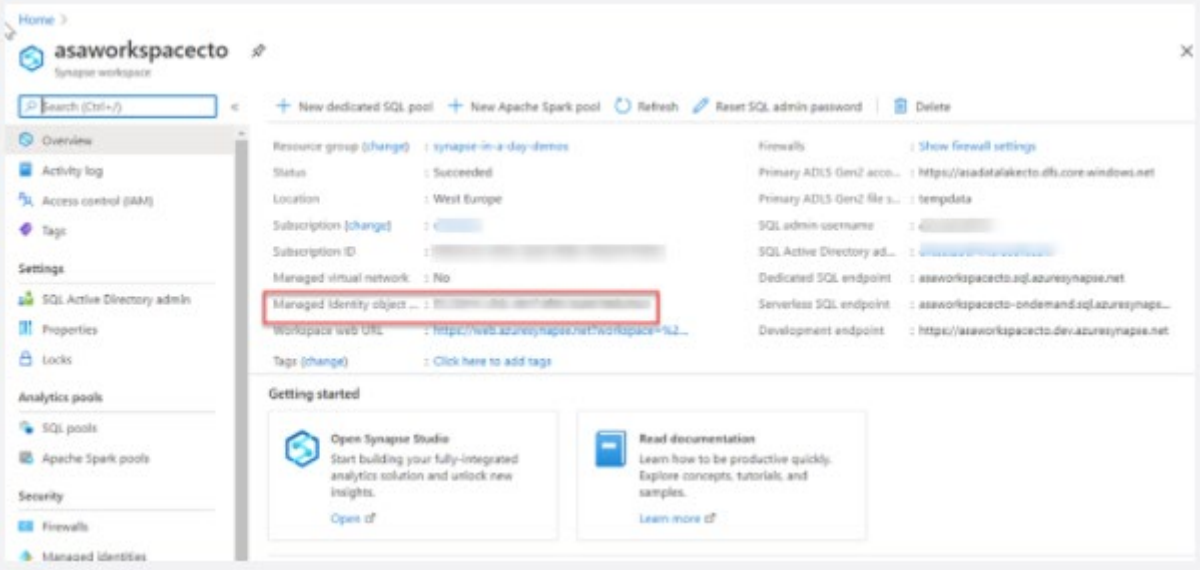
Managed identity for Azure resources is a feature of Azure Active Directory. The feature provides Azure services with an automatically managed identity in Azure AD. You can use the Managed Identity capability to authenticate to any service that support Azure Active Directory authentication.

Managed identities for Azure resources are the new name for the service formerly known as Managed Service Identity (MSI). A system-assigned managed identity is created for your Azure Synapse workspace when you create the workspace.

Azure Synapse also uses the managed identity to integrate pipelines. The managed identity lifecycle is directly tied to the Azure Synapse workspace. If you delete the Azure Synapse workspace, then the managed identity is also cleaned up.

The workspace managed identity needs permissions to perform operations in the pipelines. You can use the object ID or your Azure Synapse workspace name to find the managed identity when granting permissions.

You can retrieve the managed identity in the Azure portal. Open your Azure Synapse workspace in Azure portal and select **Overview** from the left navigation. The managed identity's object ID is displayed to in the main screen.



The managed identity information will also show up when you create a linked service that supports managed identity authentication from Azure Synapse Studio.

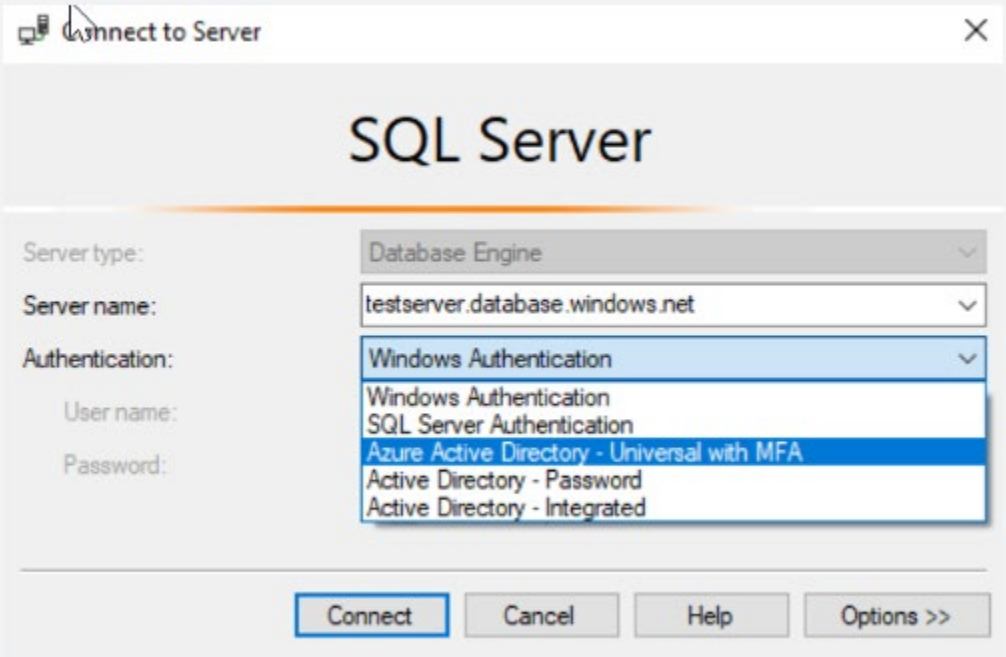
SQL Authentication

For user accounts that are not part of an Azure Active directory, then using SQL Authentication will be an alternative. In this instance, a user is created in the instance of a dedicated SQL pool. If the user in question requires administrator access, then the details of the user are held in the master database. If administrator access is not required, you can create a user in a specific database. A user then connects directly to the Azure Synapse Analytics dedicated SQL pool where they are prompted to use a username and password to access the service.

This approach is typically useful for external users who need to access the data, or if you are using third party or legacy applications against the Azure Synapse Analytics dedicated SQL pool.

Multi factor authentication

Synapse SQL support connections from SQL Server Management Studio (SSMS) using Active Directory Universal Authentication.



This enables you to operate in environments that use conditional access policies that enforce multi-factor authentication as part of the policy.

Keys

If you are unable to use a managed identity to access resources such as Azure Data Lake then you can use storage account keys and shared access signatures.

With storage account keys. Azure creates two of these keys (primary and secondary) for each storage account you create. The keys give access to everything in the account. You'll find the storage account keys in the Azure portal view of the storage account. Just select **Settings**, and then click **Access keys**.

As a best practice, you shouldn't share storage account keys, and you can use Azure Key Vault to manage and secure the keys.

Azure Key Vault is a secret store: a centralized cloud service for storing app secrets - configuration values like passwords and connection strings that must remain secure at all times. Key Vault helps you control your apps' secrets by keeping them in a single central location and providing secure access, permissions control, and access logging.

The main benefits of using Key Vault are:

• Separation of sensitive app information from other configuration and code, reducing risk of accidental leaks

• Restricted secret access with access policies tailored to the apps and individuals that need them

• Centralized secret storage, allowing required changes to happen in only one place

• Access logging and monitoring to help you understand how and when secrets are accessed

Secrets are stored in individual vaults, which are Azure resources used to group secrets together. Secret access and vault management is accomplished via a REST API, which is also supported by all of the Azure management tools as well as client libraries available for many popular languages. Every vault has a unique URL where its API is hosted.

Shared access signatures

If an external third-party application need access to your data, you'll need to secure their connections without using storage account keys. For untrusted clients, use a shared

access signature (SAS). A shared access signature is a string that contains a security token that can be attached to a URI. Use a shared access signature to delegate access to storage objects and specify constraints, such as the permissions and the time range of access. You can give a customer a shared access signature token.

Types of shared access signatures

You can use a service-level shared access signature to allow access to specific resources in a storage account. You'd use this type of shared access signature, for example, to allow an app to retrieve a list of files in a file system or to download a file.

Use an account-level shared access signature to allow access to anything that a service-level shared access signature can allow, plus additional resources and abilities. For example, you can use an account-level shared access signature to allow the ability to create file systems.

https://docs.microsoft.com/en-us/azure/synapse-analytics/security-baseline

---

Question 9: Skipped

As great as data lakes are at inexpensively storing our raw data, they also bring with them performance challenges:

• **Too many small or very big files** - more time opening & closing files rather than reading contents (worse with streaming).

• **Partitioning also known as "poor man's indexing"**- breaks down if you picked the wrong fields or when data has many dimensions, high cardinality columns.

• **No caching** - cloud storage throughput is low (cloud object storage is 20-50MB/s/core vs 300MB/s/core for local SSDs).

As a solution to the challenges with Data Lakes noted above, Delta Lake is a file format that can help you build a data lake comprised of one or many tables in Delta Lake format. Delta Lake integrates tightly with Apache Spark, and uses an open format that is based on Parquet.

Two of the core features of Delta Lake are performing `UPSERTS` and Time Travel operations.

What does the Time Travel operation do? (Select all that apply)

- ☐ Writing complex temporal queries.
  (Correct)
- ☐ Providing snapshot isolation for a set of queries for fast changing tables.
  (Correct)
- ☐ Because Delta Lake is version controlled, you have the option to query past versions of the data using a single file storage system.
  (Correct)
- ☐ Re-creating analyses, reports, or outputs (for example, the output of a machine learning model). This could be useful for debugging or auditing, especially in regulated industries.
  (Correct)

Explanation
Delta Lake is a transactional storage layer designed specifically to work with Apache Spark and Databricks File System (DBFS). At the core of Delta Lake is an optimized Spark

table. It stores your data as Apache Parquet files in DBFS and maintains a transaction log that efficiently tracks changes to the table.

Data lakes

A data lake is a storage repository that inexpensively stores a vast amount of raw data, both current and historical, in native formats such as `XML`, `JSON`, `CSV`, and `Parquet`. It may contain operational relational databases with live transactional data.

Enterprises have been spending millions of dollars getting data into data lakes with Apache Spark. The aspiration is to do data science and ML on all that data using Apache Spark.

But the data is not ready for data science & ML. The majority of these projects are failing due to unreliable data!

The challenge with data lakes

Why are these projects struggling with reliability and performance?

To extract meaningful information from a data lake, you must solve problems such as:

• Schema enforcement when new tables are introduced.

• Table repairs when any new data is inserted into the data lake.

• Frequent refreshes of metadata.

• Bottlenecks of small file sizes for distributed computations.

• Difficulty sorting data by an index if data is spread across many files and partitioned.

There are also data reliability challenges with data lakes:

• Failed production jobs leave data in corrupt state requiring tedious recovery.

• Lack of schema enforcement creates inconsistent and low quality data.

• Lack of consistency makes it almost impossible to mix appends and reads, batch and streaming.

As great as data lakes are at inexpensively storing our raw data, they also bring with them performance challenges:

• **Too many small or very big files** - more time opening & closing files rather than reading contents (worse with streaming).

• **Partitioning also known as "poor man's indexing"**- breaks down if you picked the wrong fields or when data has many dimensions, high cardinality columns.

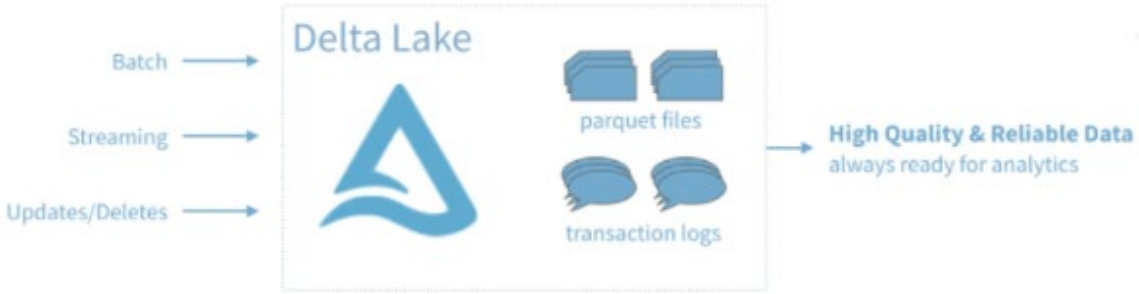
• **No caching** - cloud storage throughput is low (cloud object storage is 20-50MB/s/core vs 300MB/s/core for local SSDs).

The solution: Delta Lake

Delta Lake is a file format that can help you build a data lake comprised of one or many tables in Delta Lake format. Delta Lake integrates tightly with Apache Spark, and uses an open format that is based on Parquet. Because it is an open-source format, Delta Lake is also supported by other data platforms, including Azure Synapse Analytics.

Delta Lake makes data ready for analytics.

Delta Lake is an open-source storage layer that brings ACID transactions to Apache Spark™ and big data workloads.



You can read and write data that's stored in Delta Lake by using Apache Spark SQL batch and streaming APIs. These are the same familiar APIs that you use to work with Hive tables or DBFS directories. Delta Lake provides the following functionality:

ACID Transactions: Data lakes typically have multiple data pipelines reading and writing data concurrently, and data engineers have to go through a tedious process to ensure data integrity, due to the lack of transactions. Delta Lake brings ACID transactions to your data lakes. It provides serializability, the strongest level of isolation level.

Scalable Metadata Handling: In big data, even the metadata itself can be "big data". Delta Lake treats metadata just like data, leveraging Spark's distributed processing power to handle all its metadata. As a result, Delta Lake can handle petabyte-scale tables with billions of partitions and files at ease.

Time Travel (data versioning): Delta Lake provides snapshots of data enabling developers to access and revert to earlier versions of data for audits, rollbacks or to reproduce experiments.

Open Format: All data in Delta Lake is stored in Apache Parquet format enabling Delta Lake to leverage the efficient compression and encoding schemes that are native to Parquet.

Unified Batch and Streaming Source and Sync: A table in Delta Lake is both a batch table, as well as a streaming source and sync. Streaming data ingest, batch historic backfill, and interactive queries all just work out of the box.

Schema Enforcement: Delta Lake provides the ability to specify your schema and enforce it. This helps ensure that the data types are correct and required columns are present, preventing bad data from causing data corruption.

Schema Evolution: Big data is continuously changing. Delta Lake enables you to make changes to a table schema that can be applied automatically, without the need for cumbersome DDL.

100% Compatible with Apache Spark API: Developers can use Delta Lake with their existing data pipelines with minimal change as it is fully compatible with Spark, the commonly used big data processing engine.

Get started with Delta using Spark APIs

Delta Lake is included with Azure Databricks. You can start using it today. To quickly get started with Delta Lake, do the following:

Instead of parquet...

```
Python
CREATE TABLE ...
USING parquet
...


dataframe
.write
.format("parquet")
.save("/data")
... simply say delta
```

```
Python
CREATE TABLE ...
USING delta
...

dataframe
.write
.format("delta")
.save("/data")
```

Using Delta with your existing Parquet tables

Step 1: Convert Parquet to Delta tables:

```
Python
CONVERT TO DELTA parquet.`path/to/table` [NO STATISTICS]
[PARTITIONED BY (col_name1 col_type1, col_name2 col_type2, ...)]
Step 2: Optimize layout for fast queries:
Python
OPTIMIZE events
WHERE date >= current_timestamp() - INTERVAL 1 day
ZORDER BY (eventType)
```

Basic syntax

Two of the core features of Delta Lake are performing upserts (insert/updates) and Time Travel operations.

To `UPSERT` means to "UPdate" and "inSERT". In other words, `UPSERT` is literally TWO operations. It is not supported in traditional data lakes, as running an UPDATE could invalidate data that is accessed by the subsequent INSERT operation.

Using Delta Lake, however, we can do `UPSERTS`. Delta Lake combines these operations to guarantee atomicity to

• `INSERT` a row

• if the row already exists, `UPDATE` the row.

Upsert syntax

Upserting, or merging, in Delta Lake provides fine-grained updates of your data. The following syntax shows how to perform an Upsert:

```
SQL
MERGE INTO customers -- Delta table
USING updates
ON customers.customerId = source.customerId
WHEN MATCHED THEN
UPDATE SET address = updates.address
WHEN NOT MATCHED
THEN INSERT (customerId, address) VALUES (updates.customerId, updates.address)
```

Time Travel syntax

Because Delta Lake is version controlled, you have the option to query past versions of the data. Using a single file storage system, you now have access to several versions your historical data, ensuring that your data analysts will be able to replicate their reports (and compare aggregate changes over time) and your data scientists will be able to replicate their experiments.

Other time travel use cases are:

• Re-creating analyses, reports, or outputs (for example, the output of a machine learning model). This could be useful for debugging or auditing, especially in regulated industries.

• Writing complex temporal queries.

• Fixing mistakes in your data.

• Providing snapshot isolation for a set of queries for fast changing tables.

Example of using time travel to reproduce experiments and reports:

```
SQL
SELECT count(*) FROM events
TIMESTAMP AS OF timestamp


SELECT count(*) FROM events
```

```
VERSION AS OF version

Python

spark.read.format("delta").option("timestampAsOf", timestamp_string).load("/event
s/")

If you need to rollback accidental or bad writes:

SQL

INSERT INTO my_table

SELECT * FROM my_table TIMESTAMP AS OF

date_sub( current_date(), 1)
```

https://docs.microsoft.com/en-us/azure/synapse-analytics/spark/apache-spark-what-is-delta-lake

Question 10: Skipped

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

A(n) [?] schema may be defined at query time.

- Unstructured data type
  (Correct)
- Structured data type
- Azure Cosmos DB data type
- Hybrid data type

Explanation
An Unstructured data type schema may be defined at query time.

The schema of unstructured data is typically defined at query time. This means that data can be loaded onto a data platform in its native format.

https://docs.microsoft.com/en-us/azure/architecture/guide/technology-choices/data-store-overview

---

Question 11: Skipped

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

In Azure Data Factory, a(n) [?] is a logical grouping of activities that together perform a task.

- Sink
- Orchestration
- Linked Service
- Pipeline **(Correct)**
- Activity

Explanation

A pipeline in Azure Data Factory is a logical grouping of activities such as copy in order to perform a task. The activity defines the operation that you're performing on the data (therefore, a copy means copying the same data to another data store). For example, a pipeline could contain a set of activities that ingest and clean log data, and then kick off a mapping data flow to analyze the log data.

https://docs.microsoft.com/en-us/azure/data-factory/concepts-pipelines-activities

---

Question 12: Skipped

What does Azure Data Lake Storage (ADLS) Passthrough enable?

- Blocking ADLS resources through a mount point when credential passthrough is enabled.
- Automatically mounting ADLS accounts to the workspace that are added to the managed resource group.
- Commands running on a configured cluster can read and write data in ADLS without configuring service principal credentials.
  (Correct)
- User security groups that are added to ADLS are automatically created in the workspace as Databricks groups.

Explanation

Azure Data Lake Storage (ADLS) Passthrough enables commands running on a configured cluster can read and write data in ADLS without configuring service principal credentials. In addition, authentication to ADLS from Azure Databricks clusters is automatic, using the same Azure AD identity one uses to log into Azure Databricks.

https://docs.microsoft.com/en-us/azure/databricks/security/credential-passthrough/adls-passthrough

---

Question 13: Skipped

Azure provides many ways to store your data and there are several tools that create a storage account.

Which aspects guide a user's decision on the tool used to create a storage account? (Select two)

- ☐ The datatype being stored in the account
- ☐ Tool cost
- ☐ If the user needs automation **(Correct)**
- ☐ Location restrictions of the data centre
- ☐ If the user wants a GUI **(Correct)**

Explanation
There are several tools that create a storage account. Your choice is typically based on if you want a GUI and whether you need automation.

Available tools

The available tools are:

• Azure Portal

• Azure CLI (Command-line interface)

• Azure PowerShell

• Management client libraries

The portal provides a GUI with explanations for each setting. This makes the portal easy to use and helpful for learning about the options.

The other tools in the above list all support automation. The Azure CLI and Azure PowerShell let you write scripts, while the management libraries allow you to incorporate the creation into a client app.

How to choose a tool

Storage accounts are typically based on an analysis of your data, so they tend to be relatively stable. As a result, storage-account creation is usually a one-time operation done at the start of a project. For one-time activities, the portal is the most common choice.

In the rare cases where you need automation, the decision is between a programmatic API or a scripting solution. Scripts are typically faster to create and less work to maintain because there is no need for an IDE, NuGet packages, or build steps. If you have an existing client application, the management libraries might be an attractive choice; otherwise, scripts will likely be a better option.

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

---

Question 14: Skipped

When doing a write stream command, what does the `outputMode("append")` option do?

- The append `outputMode` allows records to update to the output log.
- The append mode allows records to be updated and changed in place.
- The append mode replaces existing records and updates aggregates.
- The append `outputMode` allows records to be added to the output sink. **(Correct)**

Explanation

The `outputMode` "append" option informs the write stream to add only new records to the output sink. The "complete" option is to rewrite the full output - applicable to aggregations operations. Finally, the "update" option is for updating changed records in place.

https://jaceklaskowski.gitbooks.io/spark-structured-streaming/content/spark-sql-streaming-MemorySink.html

---

Question 15: Skipped

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

[A] data is information that doesn't reside in a relational database but still has some structure to it. Databases that hold documents are held in [B] format.

- [A] Structured, [B] Relational
- [A] Semi-Structured, [B] JSON
  (Correct)
- [A] JSON, [B] Semi-Structured
- [A] Unstructured, [B] Binary

Explanation

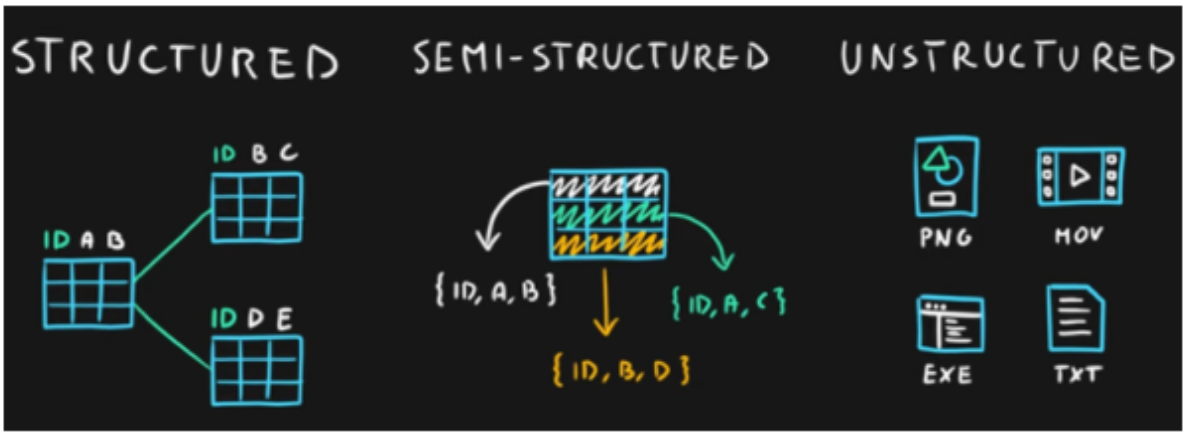
Semi-structured data is information that doesn't reside in a relational database but still has some structure to it. Examples include documents held in *JavaScript Object Notation* (JSON) format. The example below shows a pair of documents representing customer information.

```
## Document 1 ##
{
  "customerID": "103248",
  "name":
  {
    "first": "AAA",
    "last": "BBB"
  },
  "address":
  {
    "street": "Main Street",
    "number": "101",
    "city": "Acity",
    "state": "NY"
  },
  "ccOnFile": "yes",
  "firstOrder": "02/28/2003"
}
```

There are other types of semi-structured data as well. Examples include *key-value* stores and *graph* databases.

A key-value store is similar to a relational table, except that each row can have any number of columns.

You can use a graph database to store and query information about complex relationships. A graph contains nodes (information about objects), and edges (information about the relationships between objects). The image below shows an example of how you might structure the data in a graph database.



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

Question 16: Skipped

Which tool is used to perform an assessment of migrating SSIS packages to Azure SQL Database services?

- Lab Services
- Data Migration Assessment
- Data Migration Assistant
  (Correct)
- SQL Server Management Studio
- ARM templates
- Data Migration Service
- SQL Server Upgrade Advisor

Explanation

The Data Migration Assistant is used to perform an assessment of migrating SSIS packages to Azure SQL Database services.

To assess SQL Server Integration Service(SSIS) packages, below components need to be installed with Data Migration Assistant:

• SQL Server Integration Service with the same version as the SSIS packages to assess.

• Azure Feature Pack or other third party components if SSIS packages to assess have these components.

DMA needs to run with **administrator** access to assess SSIS packages in Package Store.

https://docs.microsoft.com/en-us/sql/dma/dma-assess-ssis?view=sql-server-ver15

Question 17: Skipped

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

Many business application architectures separate transactional and analytical processing into separate systems with data stored and processed on separate infrastructures. These infrastructures are commonly referred to as OLTP (online transaction processing) systems working with operational data, and OLAP (online analytical processing) systems working with historical data, with each system is optimized for their specific task.

Azure Cosmos DB provides … [?]

- None of the listed options.
- A transactional store optimized for transactional workloads and a fully managed autosync process to keep the data within these stores in sync.
- An analytical store optimized for analytical workloads and a fully managed autosync process to keep the data within these stores in sync.
- Both a transactional store optimized for transactional workloads and an analytical store optimized for analytical workloads and a fully managed autosync process to keep the data within these stores in sync.
  (Correct)

Explanation
Many business application architectures separate transactional and analytical processing into separate systems with data stored and processed on separate infrastructures. These infrastructures are commonly referred to as OLTP (online transaction processing) systems working with operational data, and OLAP (online analytical processing) systems working with historical data, with each system is optimized for their specific task.

OLTP systems are optimized for dealing with discrete system or user requests immediately and responding as quickly as possible.

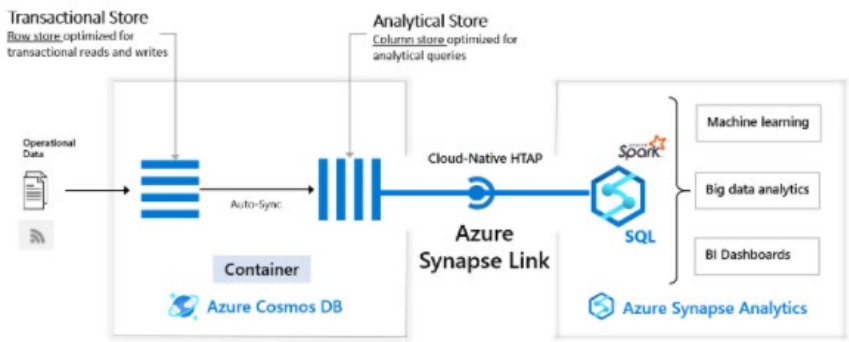
OLAP systems are optimized for the analytical processing, ingesting, synthesizing, and managing large sets of historical data. The data processed by OLAP systems largely originates from OLTP systems and needs to be loaded into the OLTP systems by means of batch processes commonly referred to as ETL (Extract, Transform, and Load) jobs.

Due to their complexity and the need to physically copy large amounts of data, this creates a delay in data being available to provide insights by way of the OLAP systems.

As more and more businesses move to digital processes, they increasingly recognize the value of being able to respond to opportunities by making faster and well-informed decisions. HTAP (Hybrid Transactional/Analytical processing) enables business to run advanced analytics in near-real-time on data stored and processed by OLTP systems.

Azure Synapse Link for Azure Cosmos DB

Azure Synapse Link for Azure Cosmos DB is a cloud-native HTAP capability that enables you to run near-real-time analytics over operational data stored in Azure Cosmos DB. Azure Synapse Link creates a tight seamless integration between Azure Cosmos DB and Azure Synapse Analytics.



Azure Cosmos DB provides both a transactional store optimized for transactional workloads and an analytical store optimized for analytical workloads and a fully managed autosync process to keep the data within these stores in sync.

Azure Synapse Analytics provides both a SQL Serverless query engine for querying the analytical store using familiar T-SQL and an Apache Spark query engine for leveraging the analytical store using your choice of Scala, Java, Python or SQL and provides a user-friendly notebook experience.

Together Azure Cosmos DB and Synapse Analytics enable organizations to generate and consume insights from their operational data in near-real time, using the query and analytics tools of their choice. All of this is achieved without the need for complex ETL pipelines and without affecting the performance of their OLTP systems using Azure Cosmos DB.

https://docs.microsoft.com/en-us/azure/cosmos-db/synapse-link

Question 18: Skipped

Scenario: A customer of Ultron Electronics is attempting to use a $300 store credit for the full amount of a new purchase. They are trying to double-spend their credit by creating two transactions at the exact same time using the entire store credit. The customer is making two transactions using two different devices.

The database behind the scenes is an ACID-compliant transactional database.

What would be the result?

- None of the listed options.
- Both orders would be processed and use the in-store credit.
- One order would be processed and use the in-store credit, and the other order would update the remaining inventory for the items in the basket, but would not complete the order.
- One order would be processed and use the in-store credit, and the other order would not be processed.
  (Correct)

Explanation
Once the second order determined that the in-store credit has already been used, it would roll back the transaction.

A transactional database must adhere to the **ACID (Atomicity, Consistency, Isolation, Durability)** properties to ensure that the database remains consistent while processing transactions.

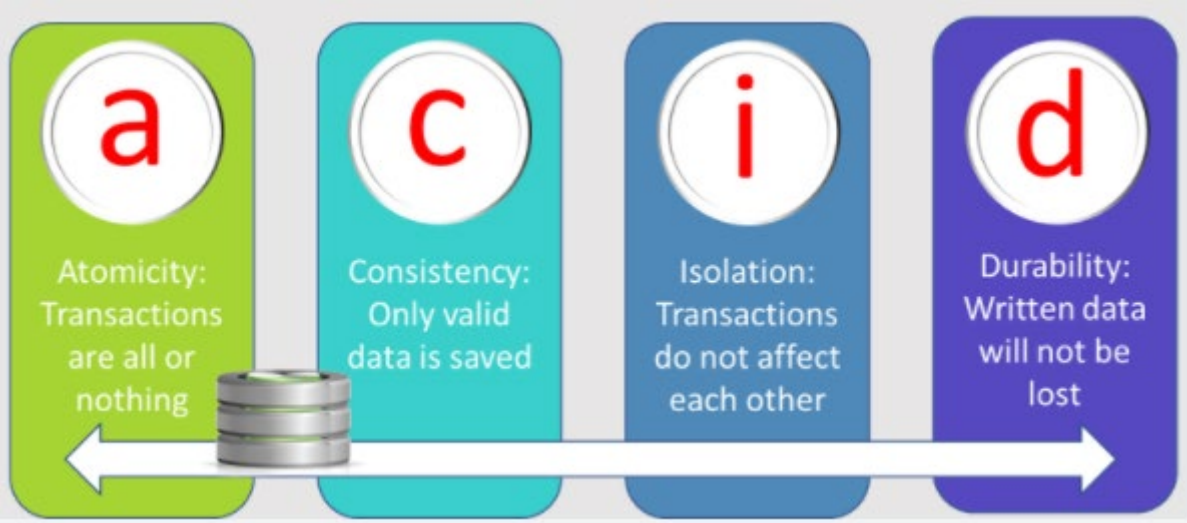
The four letters in ACID represent the four required characteristics of database transactions:

• Atomicity

• Consistency

• Isolation

• Durability



• *Atomicity* guarantees that each transaction is treated as a single *unit*, which either succeeds completely, or fails completely. If any of the statements constituting a transaction fails to complete, the entire transaction fails and the database is left unchanged. An atomic system must guarantee atomicity in each and every situation, including power failures, errors, and crashes.

• *Consistency* ensures that a transaction can only take the data in the database from one valid state to another. A consistent database should never *lose* or *create* data in a manner that can't be accounted for. In the bank transfer example described earlier, if you add funds to an account, there must be a corresponding deduction of funds somewhere, or a record that describes where the funds have come from if they have been received externally. You can't suddenly create (or lose) money.

• *Isolation* ensures that concurrent execution of transactions leaves the database in the same state that would have been obtained if the transactions were executed sequentially. A concurrent process can't see the data in an inconsistent state (for example, the funds have been deducted from one account, but not yet credited to another.)

• *Durability* guarantees that once a transaction has been committed, it will remain committed even if there's a system failure such as a power outage or crash.

https://www.techopedia.com/definition/23949/atomicity-consistency-isolation-durability-acid-database-management-system

Question 19: Skipped

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.

AIM plans to implement an Azure Cosmos DB database which will be replicated to four global regions where only the one closest to London will be writable. During events, the Cosmos DB will write 250,000 JSON each day and the consistency level must meet the following.

Requirements:

· The system must guarantee monotonic reads and writes within a session.

· The system must provide the fastest throughput available.

· Latency must be the lowest available.

As the expert, the team looks to you for direction. Which of the following consistency levels should you advise them to utilize?

- Session
  (Correct)
- Strong
- Eventual
- Consistent Prefix
- Bounded Staleness

Explanation

The key phrase is "The system must guarantee monotonic reads and writes within a session."

Consistency levels in Azure Cosmos DB

Distributed databases that rely on replication for high availability, low latency, or both, must make a fundamental tradeoff between the read consistency, availability, latency, and throughput as defined by the PACLC theorem. The linearizability of the strong consistency model is the gold standard of data programmability. But it adds a steep price from higher write latencies due to data having to replicate and commit across large distances. Strong consistency may also suffer from reduced availability (during failures) because data cannot replicate and commit in every region. Eventual consistency offers higher availability and better performance, but its more difficult to program applications because data may not be completely consistent across all regions.

Most commercially available distributed NoSQL databases available in the market today provide only strong and eventual consistency. Azure Cosmos DB offers five well-defined levels. From strongest to weakest, the levels are:

• *Strong*

• *Bounded staleness*

• *Session*

• *Consistent prefix*

• *Eventual*

Each level provides availability and performance tradeoffs. The following image shows the different consistency levels as a spectrum.

The consistency levels are region-agnostic and are guaranteed for all operations regardless of the region from which the reads and writes are served, the number of regions associated with your Azure Cosmos account, or whether your account is configured with a single or multiple write regions.

Session consistency

In session consistency, within a single client session reads are guaranteed to honor the consistent-prefix, monotonic reads, monotonic writes, read-your-writes, and write-follows-reads guarantees. This assumes a single "writer" session or sharing the session token for multiple writers.

• Clients outside of the session performing writes will see the following guarantees:

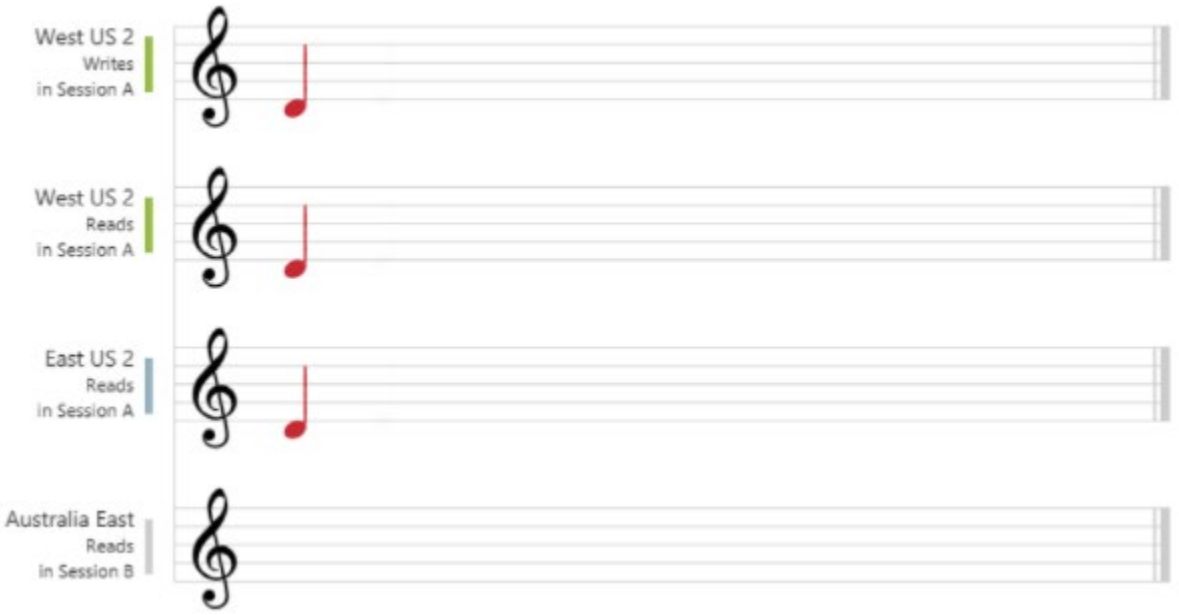
• Consistency for clients in same region for an account with single write region = Consistent Prefix

• Consistency for clients in different regions for an account with single write region = Consistent Prefix

• Consistency for clients writing to a single region for an account with multiple write regions = Consistent Prefix

• Consistency for clients writing to multiple regions for a account with multiple write regions = Eventual

Session consistency is the most widely used consistency level for both single region as well as globally distributed applications. It provides write latencies, availability, and read throughput comparable to that of eventual consistency but also provides the consistency guarantees that suit the needs of applications written to operate in the context of a user. The following graphic illustrates the session consistency with musical notes. The "West US 2 writer" and the "West US 2 reader" are using the same session (Session A) so they both read the same data at the same time. Whereas the "Australia East" region is using "Session B" so, it receives data later but in the same order as the writes.



https://docs.microsoft.com/en-us/azure/cosmos-db/consistency-levels

---

Question 20: Skipped

How can you manage the lifecycle of data and define how long it will be retained for in an analytical store?

- Configure the cache to set the time to retain the data in memory.
- Configure the default Time to Live (TTL) property for records stored. **(Correct)**
- Configure the purge duration in a container.
- Configure the deletion duration for records in the transactional store.

Explanation

Configuring the default Time to Live (TTL) property for records stored in an analytical store can manage the lifecycle of data and define how long it will be retained for.

https://help.ns1.com/hc/en-us/articles/360022250193-Best-practices-TTL-configuration

Question 21: Skipped

While Agile, CI/CD, and DevOps are different, they support one another

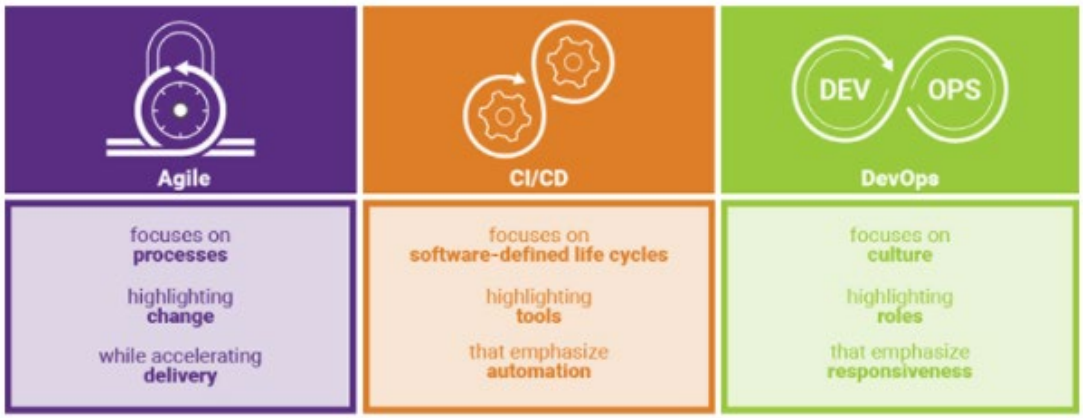
Which is best described by:

"Focuses on software-defined life cycles highlighting tools that emphasize automation."

- Agile
- DevOps
- CI/CD **(Correct)**
- SDLC

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.

• **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.

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 22: Skipped

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

The process of consuming data streams, analyzing them, and deriving actionable insights out of them is called [?].

- Event processing
  (Correct)
- Wrangling
- Multiprocessing
- Consumption

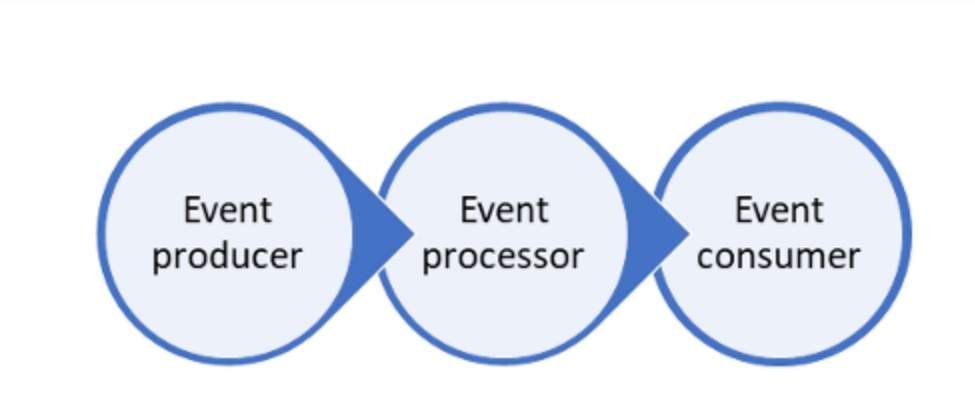
Explanation

The process of consuming data streams, analyzing them, and deriving actionable insights out of them is called event processing. An event processing pipeline has three distinct components:

• **Event producer**: Examples include sensors or processes that generate data continuously, such as a heart rate monitor or a highway toll lane sensor.

• **Event processor**: An engine to consume event data streams and derive insights from them. Depending on the problem space, event processors either process one incoming event at a time, such as a heart rate monitor, or process multiple events at a time, such as Azure Stream Analytics processing the highway toll lane sensor data.

• **Event consumer**: An application that consumes the data and takes specific action based on the insights. Examples of event consumers include alert generation, dashboards, or even sending data to another event processing engine.



https://medium.com/ek-technology/use-azure-iot-solution-build-the-first-stage-of-the-industry-4-0-44e838614f23

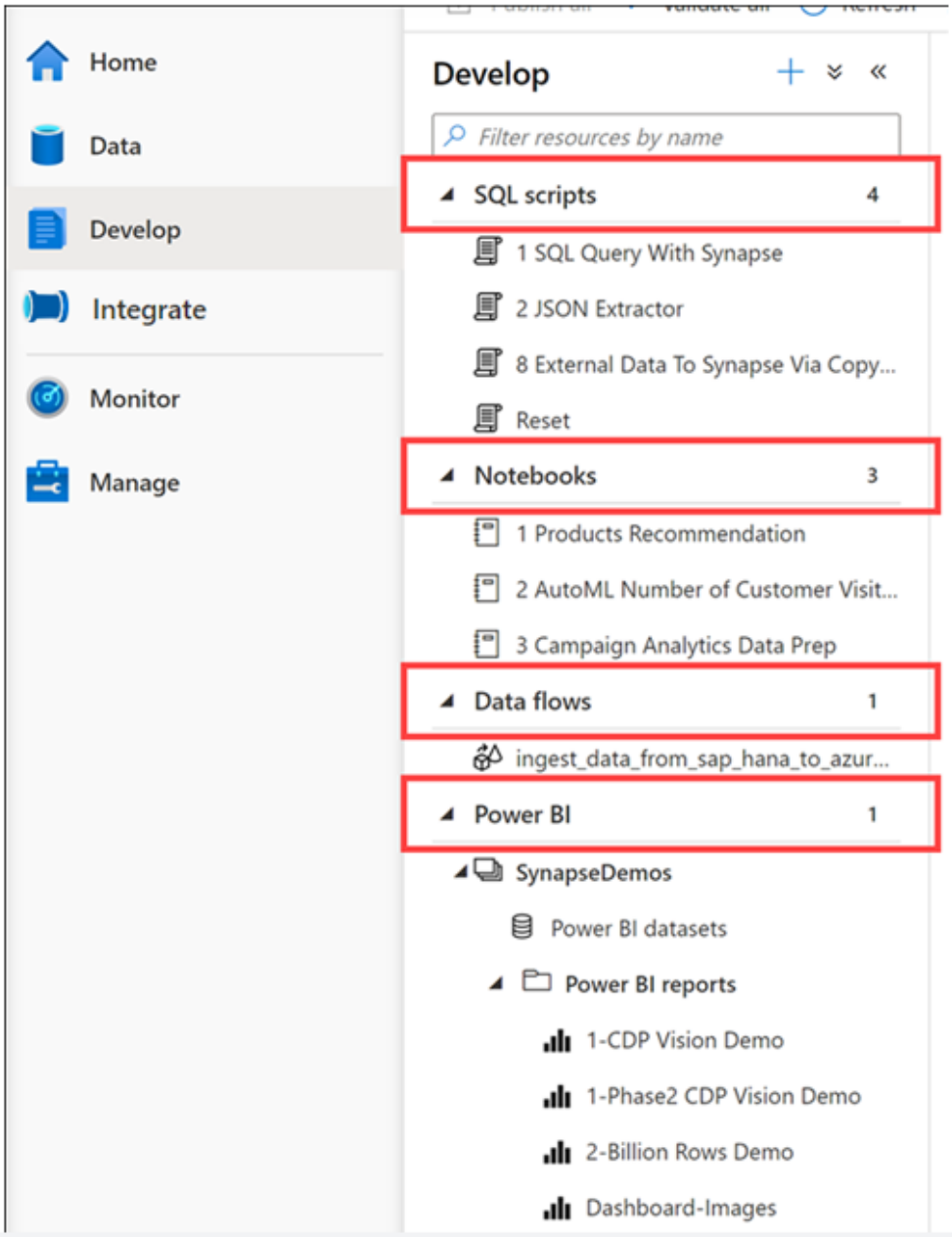
Question 23: Skipped

Which Azure Synapse Studio hub would you go to create Notebooks?

- Integrate
- Manage
- None of the listed options
  (Correct)
- Create
- Data

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 24: Skipped

Within the context of Azure Databricks, sharing data from one worker to another can be a costly operation.

Spark has optimized this operation by using a format called [?] which prevents the need for expensive serialization and de-serialization of objects in order to get data from one JVM to another.

- Tungsten
  (Correct)
- Shuffles
- Pipelining
- Stages
- Stage boundary
- Lineage

Explanation
As opposed to narrow transformations, wide transformations cause data to shuffle between executors. This is because a wide transformation requires sharing data across workers. **Pipelining** helps us optimize our operations based on the differences between the two types of transformations.

Pipelining

• Pipelining is the idea of executing as many operations as possible on a single partition of data.

• Once a single partition of data is read into RAM, Spark will combine as many narrow operations as it can into a single **Task**

• Wide operations force a shuffle, conclude a stage, and end a pipeline.

Shuffles

A shuffle operation is triggered when data needs to move between executors.

To carry out the shuffle operation Spark needs to:

• Convert the data to the `UnsafeRow`, commonly referred to as **Tungsten Binary Format**.

• Write that data to disk on the local node - at this point the slot is free for the next task.

• Send that data across the wire to another executor

• Technically the Driver decides which executor gets which piece of data.

• Then the executor pulls the data it needs from the other executor's shuffle files.

• Copy the data back into RAM on the new executor

• The concept, if not the action, is just like the initial read "every“ DataFrame starts with.

• The main difference being it's the 2nd+ stage.

As we will see in a moment, this amounts to a free cache from what is effectively temp files.

Some actions induce in a shuffle. Good examples would include the operations count() and reduce(..).

UnsafeRow (also known as Tungsten Binary Format)

Sharing data from one worker to another can be a costly operation.

Spark has optimized this operation by using a format called **Tungsten**.

Tungsten prevents the need for expensive serialization and de-serialization of objects in order to get data from one JVM to another.

The data that is "shuffled" is in a format known as `UnsafeRow`, or more commonly, the Tungsten Binary Format.

`UnsafeRow` is the in-memory storage format for Spark SQL, DataFrames & Datasets.

Advantages include:

• Compactness:

  • Column values are encoded using custom encoders, not as JVM objects (as with RDDs).

• The benefit of using Spark 2.x's custom encoders is that you get almost the same compactness as Java serialization, but significantly faster encoding/decoding speeds.

• Also, for custom data types, it is possible to write custom encoders from scratch.

• Efficiency: Spark can operate *directly out of Tungsten*, without first deserializing Tungsten data into JVM objects.

How UnsafeRow works

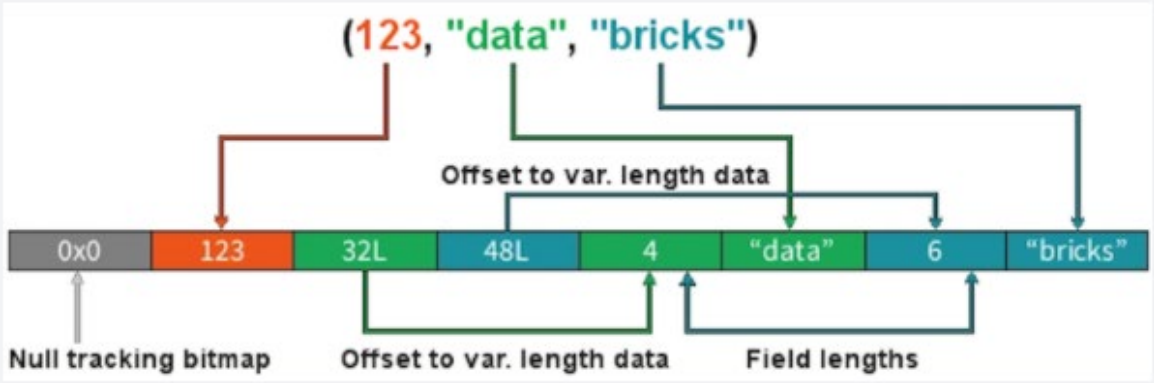
• The first field, "123", is stored in place as its primitive.

• The next 2 fields, "data" and "bricks", are strings and are of variable length.

• An offset for these two strings is stored in place (32L and 48L respectively shown in the picture below).

• The data stored in these two offset's are of format "length + data".

• At offset 32L, we store 4 + "data" and likewise at offset 48L we store 6 + "bricks".



Stages

• When we shuffle data, it creates what is known as a stage boundary.

• Stage boundaries represent a process bottleneck.

Take for example the following transformations:

Step Transformation

1 Read

2 Select

3 Filter

4 GroupBy

5 Select

6 Filter

7 Write

Spark will break this one job into two stages (steps 1-4b and steps 4c-7):

Stage #1

Step Transformation

1 Read

2 Select

3 Filter

4a GroupBy 1/2

4b shuffle write

Stage #1

Step Transformation

4c shuffle read

4d GroupBy 2/2

5 Select

6 Filter

7 Write

In **Stage #1**, Spark will create a pipeline of transformations in which the data is read into RAM (Step #1), and then perform steps #2, #3, #4a & #4b

All partitions must complete **Stage #1** before continuing to **Stage #2.**

• It's not possible to group all records across all partitions until every task is completed.

• This is the point at which all the tasks must synchronize.

• This creates our bottleneck.

• Besides the bottleneck, this is also a significant performance hit: disk IO, network IO and more disk IO.

Once the data is shuffled, we can resume execution...

For **Stage #2**, Spark will again create a pipeline of transformations in which the shuffle data is read into RAM (Step #4c) and then perform transformations #4d, #5, #6 and finally the write action, step #7.

Lineage

From the developer's perspective, we start with a read and conclude (in this case) with a write:

Step Transformation

1 Read

2 Select

3 Filter

4 GroupBy

5 Select

6 Filter

7 Write

However, Spark starts with the action (write(..) in this case).

Next, it asks the question, what do I need to do first?

It then proceeds to determine which transformation precedes this step until it identifies the first transformation.

Step Transformation

7 Write Depends on #6

6 Filter Depends on #5

5 Select Depends on #4

4 GroupBy Depends on #3

3 Filter Depends on #2

2 Select Depends on #1

1 Read First

Why Work Backwards?

Question: So what is the benefit of working backward through your action's lineage?

Answer: It allows Spark to determine if it is necessary to execute every transformation.

Take another look at our example:

• Say we've executed this once already

• On the first execution, step #4 resulted in a shuffle

• Those shuffle files are on the various executors (src & dst)

• Because the transformations are immutable, no aspect of our lineage can change.

• That means the results of our last shuffle (if still available) can be reused.

Step Transformation

7 Write Depends on #6

6 Filter Depends on #5

5 Select Depends on #4

4 GroupBy <<< shuffle

3 Filter don't care

2 Select don't care

1 Read don't care

In this case, what we end up executing is only the operations from **Stage #2**.

This saves us the initial network read and all the transformations in **Stage #1**

Step Transformation

1 Read skipped

2 Select skipped

3 Filter skipped

4a GroupBy 1/2 skipped

4b shuffle write skipped

4c shuffle read -

4d GroupBy 2/2 -

5 Select -

6 Filter -

7 Write

And Caching...

The reuse of shuffle files (also known as our temp files) is just one example of Spark optimizing queries anywhere it can.

We cannot assume this will be available to us.

Shuffle files are by definition temporary files and will eventually be removed.

However, we cache data to explicitly accomplish the same thing that happens inadvertently with shuffle files.

In this case, the lineage plays the same role. Take for example:

Step Transformation

7 Write Depends on #6

6 Filter Depends on #5

5 Select <<< cache

4 GroupBy <<< shuffle files

3 Filter ?

2 Select ?

1 Read ?

In this case we cached the result of the select(..).

We never even get to the part of the lineage that involves the shuffle, let alone Stage #1.

Instead, we pick up with the cache and resume execution from there:

Step Transformation

1 Read skipped

2 Select skipped

3 Filter skipped

4a GroupBy 1/2 skipped

4b shuffle write skipped

4c shuffle read skipped

4d GroupBy 2/2 skipped

5a cache read -

5b Select -

6 Filter -

7 Write

https://databricks.com/blog/2015/04/28/project-tungsten-bringing-spark-closer-to-bare-metal.html

---

Question 25: Skipped

One of the key management features that you have at your disposal within Azure Synapse Analytics, is the ability the scale the compute resources for SQL or Spark pools to meet the demands of processing your data. Compute is separate from storage, which enables you to scale compute independently of the data in your system. This means you can scale up and scale down the compute power to meet your needs.

Apache Spark pools for Azure Synapse Analytics uses an Autoscale feature that automatically scales the number of nodes in a cluster instance up and down.

Autoscale continuously monitors the Spark instance and collects which of the following metrics? (Select five)

- ☐ Total Free Memory **(Correct)**
- ☐ Total Free CPU **(Correct)**
- ☐ Average Refresh rate
- ☐ Total Pending CPU **(Correct)**
- ☐ Total seeds on each individual Node
- ☐ Used Memory per Node **(Correct)**
- ☐ Total seeds on the Node collective
- ☐ Total number of peers on the Node network
- ☐ Total Pending Memory **(Correct)**

Explanation
One of the key management features that you have at your disposal within Azure Synapse Analytics, is the ability the scale the compute resources for SQL or Spark pools to meet the demands of processing your data. In SQL pools, the unit of scale is an abstraction of

compute power that is known as a data warehouse unit. Compute is separate from storage, which enables you to scale compute independently of the data in your system. This means you can scale up and scale down the compute power to meet your needs.

You can scale a Synapse SQL pool either through the Azure portal, Azure Synapse Studio or programmatically using T-SQL or PowerShell.

Scaling Apache Spark pools in Azure Synapse Analytics

Apache Spark pools for Azure Synapse Analytics uses an **Autoscale** feature that automatically scales the number of nodes in a cluster instance up and down. During the creation of a new Spark pool, a minimum and maximum number of nodes can be set when **Autoscale** is selected. Autoscale then monitors the resource requirements of the load and scales the number of nodes up or down. 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.

Autoscale continuously monitors the Spark instance and collects the following metrics:

• 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 following conditions will then autoscale the memory or CPU

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.

The scaling operation can take between 1 -5 minutes. During an instance where there is a scale down process, Autoscale will put the nodes in decommissioning state so that no new executors can launch on that node.

The running jobs will continue to run and finish. The pending jobs will wait to be scheduled as normal with fewer available nodes.

https://docs.microsoft.com/en-us/azure/synapse-analytics/sql-data-warehouse/quickstart-scale-compute-portal

---

Question 26: Skipped

Azure Storage accounts are the base storage type within Azure. Azure Storage offers a very scalable object store for data objects and file system services in the cloud. It can also provide a messaging store for reliable messaging, or it can act as a NoSQL store.

Which of the following are Azure Storage configuration options? (Select all that apply)

- Azure Blob
(Correct)
- Azure Cosmos DB
- Azure Queue
(Correct)
- Azure Data Lake
(Correct)
- Azure Table
(Correct)
- Azure Database Server
- Azure Files
(Correct)

Explanation
Azure Storage accounts are the base storage type within Azure. Azure Storage offers a very scalable object store for data objects and file system services in the cloud. It can also provide a messaging store for reliable messaging, or it can act as a NoSQL store.

Azure Storage offers four configuration options:

• **Azure Blob**: A scalable object store for text and binary data

• **Azure Files**: Managed file shares for cloud or on-premises deployments

• **Azure Queue**: A messaging store for reliable messaging between application components

• **Azure Table**: A NoSQL store for no-schema storage of structured data

You can use Azure Storage as the storage basis when you're provisioning a data platform technology such as Azure Data Lake Storage and HDInsight. But you can also provision Azure Storage for standalone use. For example, you provision an Azure Blob store either as standard storage in the form of magnetic disk storage or as premium storage in the form of solid-state drives (SSDs).

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

• **Azure Data Lake:** Microsoft Azure Data Lake is a technology in Azure cloud that enables big data analytics and artificial intelligence (AI). When this topic mentions "Data Lake," it's referring specifically to storage technology that is based on Azure Data Lake Storage Gen2.

https://docs.microsoft.com/en-us/dynamics365/fin-ops-core/dev-itpro/data-entities/azure-data-lake-overview

---

Question 27: Skipped

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

[?] provides one-click setup, streamlined workflows, an interactive workspace for Spark-based applications plus it adds capabilities to Apache Spark, including fully managed Spark clusters and an interactive workspace.

- Azure Databricks
  (Correct)
- Azure SQL Datawarehouse
- Azure Data Factory
- Azure Data Catalogue
- Azure Cosmos DB
- Azure Storage Explorer
- Azure Data Lake Storage

Explanation
Azure Databricks

Databricks is a serverless platform that's optimized for Azure. It provides one-click setup, streamlined workflows, and an interactive workspace for Spark-based applications.

Databricks adds capabilities to Apache Spark, including fully managed Spark clusters and an interactive workspace. You can use REST APIs to program clusters.

In Databricks notebooks you'll use familiar programming tools such as R, Python, Scala, and SQL. Role-based security in Azure Active Directory and Databricks provides enterprise-grade security.

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

---

Question 28: Skipped

Azure offers several types of storage for data, the one chosen should depend on the needs of the users. Each data store has a different price structure. When you want to store data but don't need to query it, which would be the most cost efficient choice?

- Azure Storage
(Correct)

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

Explanation
Azure Storage offers a massively scalable object store for data objects and file system services for the cloud. There are several options for ingesting data into Azure, depending on your needs.

File storage:

• Azure Storage blobs

• Azure Data Lake Store

NoSQL databases:

• Azure Cosmos DB

• HBase on HDInsight

Analytical databases:

• Azure Data Explorer

If you create a Blob storage account, you can't directly query the data.

https://docs.microsoft.com/en-us/azure/architecture/data-guide/technology-choices/data-storage

Question 29: Skipped

During the process of creating a notebook, you need to specify the pool that needs to be attached to the notebook that is, a SQL or Spark pool.

True or False: Notebook cells are individual blocks of code or text that runs as a group. If you want to skip cells within the group, a simple skip notation in the cell is all that is required.

- True
- False **(Correct)**

Explanation

When you want to run notebooks in the Synapse Studio environment, you are able to run the code cells in your notebook individually or all at once. The status and progress of each cell will also be represented in the notebook.

The different functionalities for running a notebook are as follows:

• Run a Cell If you want to run once cell of code or text, you can do so through the notebook experience in Azure Synapse Studio.

• Run all cells. If you have developed code that consists of multiple cells or is combined with text, you can do so through the notebook experience in Azure Synapse Studio.

• Cancel a running cell. If you hit run, but while the cell is running, you want to cancel one cell run, you can do so within Azure Synapse notebooks.

• Cell Status indicator If you want to check the status of a cell while running or completed, you have the possibility to get a status indicator within the notebook experience in Synapse Studio.

• Spark progress indicator Azure Synapse Studio notebook is purely Spark based. Remotely, the code cells that are executed, are executed on the serverless Apache Spark pool. If you want to see the progress of a spark job, you can see in real time the job execution status below a cell. The number of tasks per each job or stage help you to identify the parallel level of your spark job. You can also drill deeper to the Spark UI of a specific job (or stage) via selecting the link on the job (or stage) name.

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

Question 30: Skipped

Azure Synapse Analytics is a high performing Massively Parallel Processing (MPP) engine that is built with loading and querying large datasets in mind.

There are times though when performance expectations are not met, and it is necessary then to know what aspects of the table structures and architecture can be reviewed and adapted to maximize query performance.

What is the following code intended to accomplish?

```
1. PowerShell
2. Set-AzSqlDatabase -ResourceGroupName "resourcegroupname" -DatabaseName "mySampleDataWar
   ehouse" -ServerName "sqlpoolservername" -RequestedServiceObjectiveName "DW300c"
```

- Address the issue of low concurrency.
  (Correct)
- None of the listed options.
- Address the issue of poor query performance.
- Address the issue of poor response time.
- Address the issue of poor load performance.

Explanation

Azure Synapse Analytics is a high performing Massively Parallel Processing (MPP) engine that is built with loading and querying large datasets in mind. Many query performance enhancements are enabled by default for querying data from Azure Synapse Analytics, and additional capabilities and enhancements have both been inherited from the SQL Server product family, and have features also designed specifically to leverage the MPP capabilities within the dedicated SQL Pools architecture.

There are times though when performance expectations are not met, and it is necessary then to know what aspects of the table structures and architecture can be reviewed and adapted to maximize query performance. Symptoms that indicate that there are performance issues related to tables include:

Poor query performance

The first indication of a poor query performance issue is typically from business users who may report that their business reports are slow, or sometime not even appearing.

Poor load performance

Poor load performance may be reported by telemetry of the data loads through Azure Synapse pipelines, or you may get users reporting that the data in the reports is out of date.

Low concurrency

You may receive reports from your users that they may be unable to connect to the data warehouse to execute reports or queries.

The first response will be to ensure that the data warehouse is set to the appropriate service level range to ensure there is enough memory and concurrency slots available for multiple connections to the service. Scaling the service within the Azure portal, or Azure Synapse Studio, or issuing a Transact-SQL or the following PowerShell statement will address the issue of low concurrency.

```
PowerShell

Set-AzSqlDatabase -ResourceGroupName "resourcegroupname" -DatabaseName "mySampleDataWarehouse" -ServerName "sqlpoolservername" -RequestedServiceObjectiveName "DW300c"
```

Even with these changes, performance issue may not be resolved. Then you would have to explore other areas that we will explore in this module to resolve the issue.

https://docs.microsoft.com/en-us/azure/databricks/data/data-sources/azure/synapse-analytics

---

Question 31: Skipped

Azure offers a service to detect anomalies in account activities. These anomalies generate security alerts which are integrated with Azure Security Centre, and are also sent via email to subscription administrators, with details of suspicious activity and recommendations on how to investigate and remediate threats.

Which of the below is the name of this service?

- Azure Armour for Storage
- Azure Shield for Storage
- Encryption in transit
- Azure Storage Account Security Feature
- Azure Defender for Storage
  (Correct)

Explanation
Microsoft Defender for Storage detects anomalies in account activity. It then notifies you of potentially harmful attempts to access your account.

Azure Defender for Storage provides an extra layer of security intelligence that detects unusual and potentially harmful attempts to access or exploit storage accounts. This layer of protection allows you to address threats without being a security expert or managing security monitoring systems.

Security alerts are triggered when anomalies in activity occur. These security alerts are integrated with Azure Security Centre, and are also sent via email to subscription administrators, with details of suspicious activity and recommendations on how to investigate and remediate threats.

Azure Defender for Storage is currently available for Blob storage, Azure Files, and Azure Data Lake Storage Gen2. Account types that support Azure Defender include general-purpose v2, block blob, and Blob storage accounts. Azure Defender for Storage is available in all public clouds and US government clouds, but not in other sovereign or Azure Government cloud regions.

Accounts with hierarchical namespaces enabled for Data Lake Storage support transactions using both the Azure Blob storage APIs and the Data Lake Storage APIs. Azure file shares support transactions over SMB.

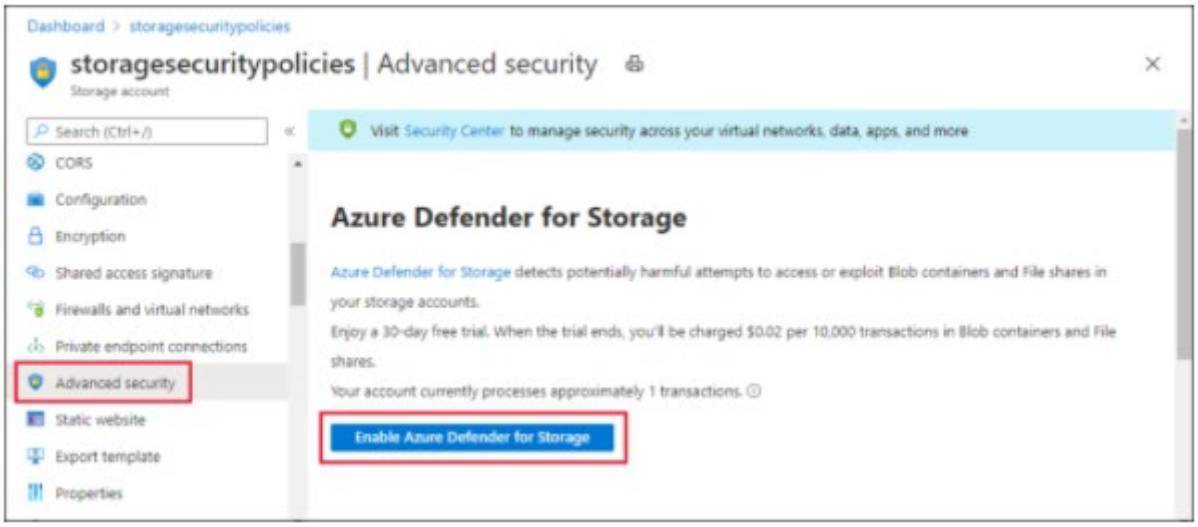
You can turn on Azure Defender for Storage in the Azure portal through the configuration page of the Azure Storage account, or in the advanced security section of the Azure portal.

Follow these steps.

1. Launch the Azure portal.

2. Navigate to your storage account. Under **Settings**, select **Advanced security**.

3. Select **Enable Azure Defender for Storage**.



https://docs.microsoft.com/en-us/azure/security-center/defender-for-storage-introduction

Question 32: Skipped

Which DataFrame method do you use to create a temporary view?

- `createOrReplaceTempView()`
(Correct)
- `tempViewCreate()`
- `createTempViewDF()`
- `createTempView()`

Explanation
You use this method to create temporary views in DataFrames.

CREATE VIEW

Constructs a virtual table that has no physical data based on the result-set of a SQL query. `ALTER VIEW` and `DROP VIEW` only change metadata.

Syntax

```
SQL

CREATE [ OR REPLACE ] [ [ GLOBAL ] TEMPORARY ] VIEW [ IF NOT EXISTS ] view_identifier
```

`create_view_clauses AS query`

Parameters

• `OR REPLACE`

If a view of same name already exists, it is replaced.

• `[ GLOBAL ] TEMPORARY`

`TEMPORARY` views are session-scoped and is dropped when session ends because it skips persisting the definition in the underlying metastore, if any. `GLOBAL TEMPORARY` views are tied to a system preserved temporary database `global_temp`.

• `IF NOT EXISTS`

Creates a view if it does not exist.

• `view_identifier`

A view name, optionally qualified with a database name.

Syntax: `[database_name.] view_name`

• create_view_clauses

These clauses are optional and order insensitive. It can be of following formats.

• `[ ( column_name [ COMMENT column_comment ], ... ) ]` to specify column-level comments.

• `[ COMMENT view_comment ]` to specify view-level comments.

• `[ TBLPROPERTIES ( property_name = property_value [ , ... ] ) ]` to add metadata key-value pairs.

• query a `SELECT` statement that constructs the view from base tables or other views.

```
SQL
-- Create or replace view for `experienced_employee` with comments.
CREATE OR REPLACE VIEW experienced_employee
(ID COMMENT 'Unique identification number', Name)
COMMENT 'View for experienced employees'
AS SELECT id, name FROM all_employee
WHERE working_years > 5;


-- Create a global temporary view `subscribed_movies` if it does not exist.
CREATE GLOBAL TEMPORARY VIEW IF NOT EXISTS subscribed_movies
AS SELECT mo.member_id, mb.full_name, mo.movie_title
FROM movies AS mo INNER JOIN members AS mb
ON mo.member_id = mb.id;
```

https://docs.databricks.com/spark/latest/spark-sql/language-manual/sql-ref-syntax-ddl-create-view.html

Question 33: Skipped

How do you disable Azure Synapse Link for Azure Cosmos DB?

- Delete the Azure Cosmos DB container
- Delete the Azure Cosmos DB account
  (Correct)
- Set the Azure Synapse Link option to disable on the Azure Cosmos DB instance.
- Set the Azure Synapse Link option to disable on the Azure Cosmos DB container.

Explanation

After the Synapse Link capability is enabled at the account level, you cannot disable it. Understand that you will not have any billing implications if the Synapse Link capability is enabled at the account level and there is no analytical store enabled containers.

If you need to turn off the capability, you have 2 options.

• The first one is to delete and re-create a new Azure Cosmos DB account, migrating the data if necessary.

• The second option is to open a support ticket, to get help on a data migration to another account.

Deleting the Azure Cosmos DB account with disable and remove Azure Synapse Link.

https://docs.microsoft.com/en-us/azure/cosmos-db/synapse-link-frequently-asked-questions

---

Question 34: Skipped

Azure Synapse SQL pools support placing complex data processing logic into Stored procedures

True or False: Multiple users and client programs can perform operations on underlying database objects through a procedure, even if the users and programs do not have direct permissions on those underlying objects.

- True
  (Correct)
- False

Explanation

Azure Synapse SQL pools support placing complex data processing logic into Stored procedures. Stored procedures are great way of encapsulating one or more SQL statements or a reference to a Microsoft .NET framework Common Language Runtime (CLR) method.

Stored procedures can accept input parameters and return multiple values in the form of output parameters to the calling program. In the context of serverless SQL pools, you will perform data transformation using `CREATE EXTERNAL TABLE AS SELECT (CETAS)` statement as shown in the following example.

```
SQL

-- this sample references external data source and external file format defined i
n previous section

CREATE PROCEDURE usp_calculate_population_by_year_state

AS

BEGIN

CREATE EXTERNAL TABLE population_by_year_state

WITH (

LOCATION = 'population_by_year_state/',

DATA_SOURCE = destination_ds,

FILE_FORMAT = parquet_file_format

)

AS

SELECT decennialTime, stateName, SUM(population) AS population
```

```
FROM

OPENROWSET(BULK 'https://azureopendatastorage.blob.core.windows.net/censusdatacon
tainer/release/us_population_county/year=*/*.parquet',

FORMAT='PARQUET') AS [r]

GROUP BY decennialTime, stateName

END

GO
```

In addition to encapsulating Transact-SQL logic, stored procedures also provide the following additional benefits:

Reduces client to server network traffic

The commands in a procedure are executed as a single batch of code. This can significantly reduce network traffic between the server and client because only the call to execute the procedure is sent across the network.

Provides a security boundary

Multiple users and client programs can perform operations on underlying database objects through a procedure, even if the users and programs do not have direct permissions on those underlying objects. The procedure controls what processes and activities are performed and protects the underlying database objects. This eliminates the requirement to grant permissions at the individual object level and simplifies the security layers.

Eases maintenance

When client applications call procedures and keep database operations in the data tier, only the procedures must be updated for any changes in the underlying database.

Improved performance

Stored procedures are compiled the first time they are executed, and the subsequent execution plan is held in the cache and reused on subsequent execution of the same stored procedure. As a result, it takes less time to process the procedure.

https://docs.microsoft.com/en-us/azure/synapse-analytics/sql/develop-stored-procedures

---

Question 35: Skipped

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 if you do not have an analytical environment in place already.

[?] is a single web UI that allows you to:

• Explore your data estate.

• Develop TSQL scripts and notebooks to interact with the analytical engines.

• Build data integration pipelines for managing data movement.

• Monitor the workloads within the service.

• Manage the components of the service.

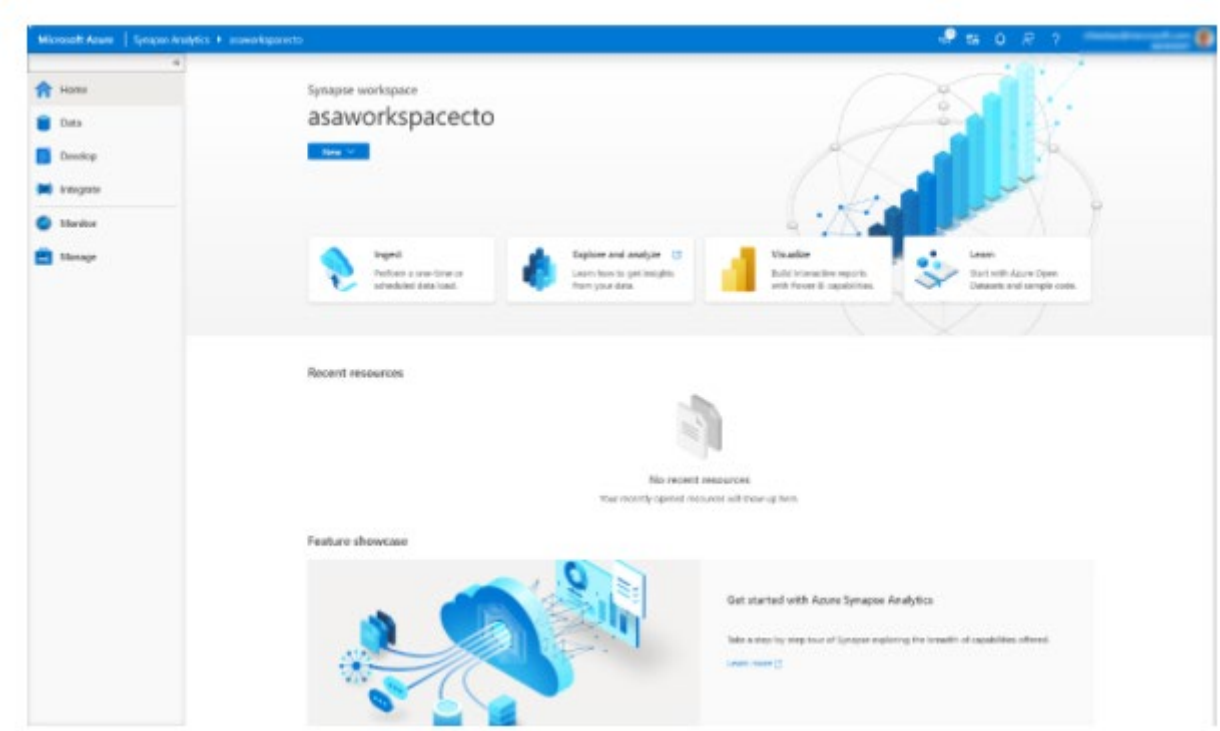
- Azure Synapse Studio
(Correct)
- Azure Portal
- Azure Monitor
- Azure Designer
- Azure Pipelines
- Azure DevOps

Explanation

Azure Synapse Analytics can work by acting as the one stop shop to meet all of your analytical needs in an integrated environment if you do not have an analytical environment in place already.

A single Web UI to be able to access all Azure Synapse Analytics capabilities

While the Azure Portal will allow you to manage some aspects of the product, Azure Synapse Studio is the best place to centrally work with all the capabilities.



Azure Synapse Studio is a single web UI that allows you to:

• Explore your data estate.

• Develop TSQL scripts and notebooks to interact with the analytical engines.

• Build data integration pipelines for managing data movement.

• Monitor the workloads within the service.

• Manage the components of the service.

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

---

Question 36: Skipped

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

As a Data Engineer, you can transfer and move data in several ways. The most common tool is Azure Data Factory which provides robust resources and nearly 100 enterprise connectors. Azure Data Factory also allows you to transform data by using a wide variety of languages.

Azure has opened the way for technologies that can handle unstructured data at an unlimited scale. This change has shifted the paradigm for loading and transforming data from [?].

- ETL → MTD
- ETL → RTO
- ETL → ELT
  (Correct)
- RPO → RTO
- ELT → ETL
- MTD → RPO

Explanation

As a Data Engineer, you can transfer and move data in several ways. One way is to start an *Extract, Transform, and Load (ETL)* process.

Extraction sources can include databases, files, and streams. Each source has unique data formats that can be structured, semistructured, or unstructured. In Azure, data sources include Azure Cosmos DB, Azure Data Lake, files, and Azure Blob storage.

ETL tools

As a data engineer, you'll use several tools for ETL. The most common tool is Azure Data Factory, which provides robust resources and nearly 100 enterprise connectors. Data Factory also allows you to transform data by using a wide variety of languages.

You might find that you also need a repository to maintain information about your organization's data sources and dictionaries. Azure Data Catalogue can store this information centrally.

Azure Data Factory

Data Factory is a cloud-integration service. It orchestrates the movement of data between various data stores.

As a data engineer, you can create data-driven workflows in the cloud to orchestrate and automate data movement and data transformation. Use Data Factory to create and schedule data-driven workflows (called pipelines) that can ingest data from data stores.

Data Factory processes and transforms data by using compute services such as Azure HDInsight, Hadoop, Spark, and Azure Machine Learning. Publish output data to data stores such as Azure SQL Data Warehouse so that business intelligence applications can consume the data. Ultimately, you use Data Factory to organize raw data into meaningful data stores and data lakes so your organization can make better business decisions.

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

Evolution from ETL

Azure has opened the way for technologies that can handle unstructured data at an unlimited scale. This change has shifted the paradigm for loading and transforming data from ETL to extract, load, and transform (ELT).

The benefit of ELT is that you can store data in its original format, be it JSON, XML, PDF, or images. In ELT, you define the data's structure during the transformation phase, so you can use the source data in multiple downstream systems.

In an ELT process, data is extracted and loaded in its native format. This change reduces the time required to load the data into a destination system. The change also limits resource contention on the data sources.

The steps for the ELT process are the same as for the ETL process. They just follow a different order.

Another process like ELT is called extract, load, transform, and load (ELTL). The difference with ELTL is that it has a final load into a destination system.

https://docs.microsoft.com/en-us/azure/architecture/data-guide/relational-data/etl

Question 37: Skipped

What type of process are the driver and the executors?

- JavaScript
- C++ processes
- Python processes
- Java processes
  (Correct)

Explanation
The driver and the executors are Java processes.

What is a JVM?

The JVM manages system memory and provides a portable execution environment for Java-based applications

Technical definition: The JVM is the specification for a software program that executes code and provides the runtime environment for that code.

Everyday definition: The JVM is how we run our Java programs. We configure the JVM's settings and then rely on it to manage program resources during execution.

The **Java Virtual Machine (JVM)** is a program whose purpose is to execute other programs.

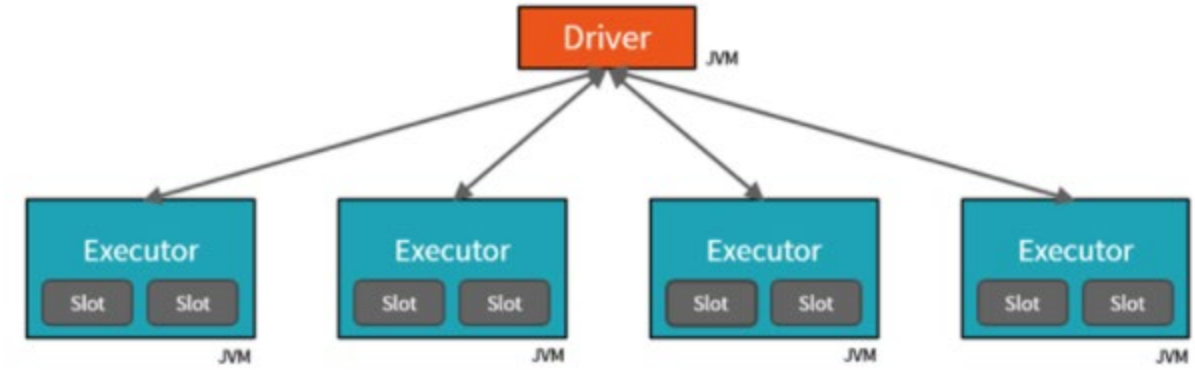
The JVM has **two primary functions**:

1.To allow Java programs to run on any device or operating system (known as the *"Write once, run anywhere"* principle)

2.To manage and optimize program memory

JVM view of the Spark Cluster: *Drivers, Executors, Slots & Tasks*

The Spark runtime architecture leverages JVMs:



https://www.rakirahman.me/spark-certification-study-guide-part-1/

Question 38: Skipped

Scenario: Queen Consolidated was overtaken by Raymond Carson Palmer and re-branded as Palmer Technologies. Now that Ray is overseeing the operations at Palmer, Ray has decided to implement better applications.

You are working as a consultant with Palmer, and in a meeting with Ray and his IT team discussing Azure Data Factory. The team plans to use Azure Data Factory to prepare data to be queried by Azure Synapse Analytics serverless SQL pools.

Files will be initially ingested into an Azure Data Lake Storage Gen2 account as 10 small JSON files. Each file will contain the same data attributes and data from a subsidiary of Palmer. The team needs to move the files to a different folder and transform the data.

Required:
• Provide the fastest possible query times.
• Automatically infer the schema from the underlying files.

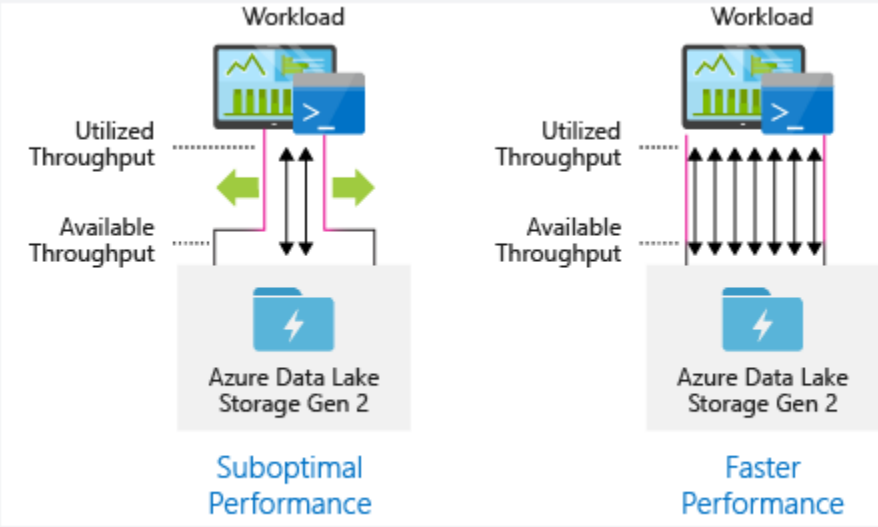
As the Azure expert, the team looks to you for advice on how they should configure the Data Factory copy activity with respect to the copy behaviour.

Which of the following should you advise them to use?

- Preserve hierarchy
- Append Files
- Flatten hierarchy
- Merge Files
  (Correct)

Explanation
With respect to the copy behaviour, you should advise the team to Merge files.

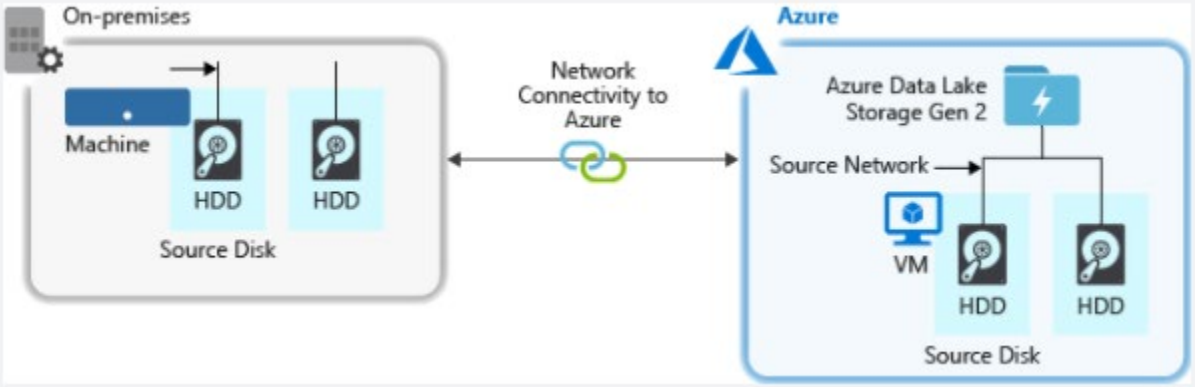
Azure Data Lake Storage Gen2 supports high-throughput for I/O intensive analytics and data movement. In Data Lake Storage Gen2, using all available throughput – the amount of data that can be read or written per second – is important to get the best performance. This is achieved by performing as many reads and writes in parallel as possible.



Data Lake Storage Gen2 can scale to provide the necessary throughput for all analytics scenarios. By default, a Data Lake Storage Gen2 account provides enough throughput in its default configuration to meet the needs of a broad category of use cases. For the cases where customers run into the default limit, the Data Lake Storage Gen2 account can be configured to provide more throughput by contacting Azure Support.

Data ingestion

When ingesting data from a source system to Data Lake Storage Gen2, it is important to consider that the source hardware, source network hardware, or network connectivity to Data Lake Storage Gen2 can be the bottleneck.



It is important to ensure that the data movement is not affected by these factors.

Optimizing I/O intensive jobs on Hadoop and Spark workloads on HDInsight

Jobs fall into one of the following three categories:

CPU intensive. These jobs have long computation times with minimal I/O times. Examples include machine learning and natural language processing jobs.

Memory intensive. These jobs use lots of memory. Examples include PageRank and real-time analytics jobs.

I/O intensive. These jobs spend most of their time doing I/O. A common example is a copy job which does only read and write operations. Other examples include data preparation jobs that read a lot of data, performs some data transformation, and then writes the data back to the store.

The following guidance is only applicable to I/O intensive jobs.

General considerations

You can have a job that reads or writes as much as 100MB in a single operation, but a buffer of that size might compromise performance. To optimize performance, try to keep the size of an I/O operation between 4MB and 16MB.

General considerations for an HDInsight cluster

HDInsight versions. For best performance, use the latest release of HDInsight.

Regions. Place the Data Lake Storage Gen2 account in the same region as the HDInsight cluster.

An HDInsight cluster is composed of two head nodes and some worker nodes. Each worker node provides a specific number of cores and memory, which is determined by the VM-type. When running a job, YARN is the resource negotiator that allocates the available memory and cores to create containers. Each container runs the tasks needed to complete the job. Containers run in parallel to process tasks quickly. Therefore, performance is improved by running as many parallel containers as possible.

There are three layers within an HDInsight cluster that can be tuned to increase the number of containers and use all available throughput.

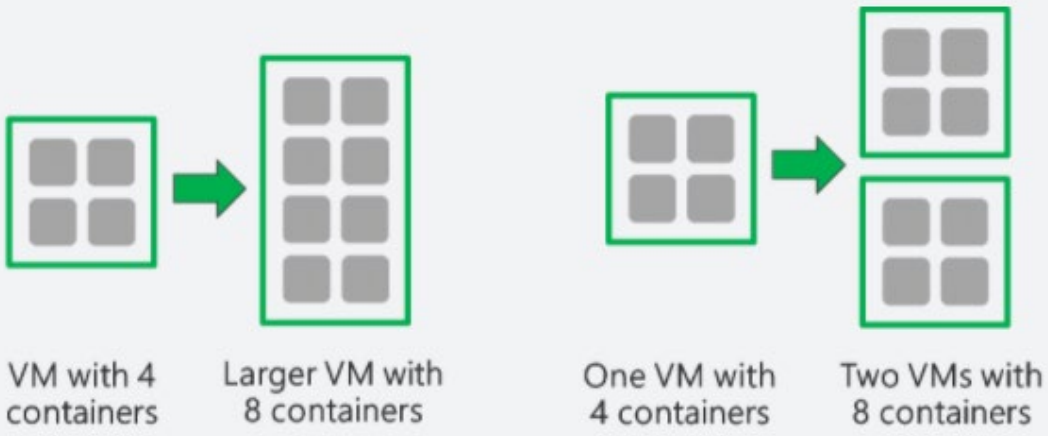
Physical layer

YARN layer

Workload layer

Physical Layer

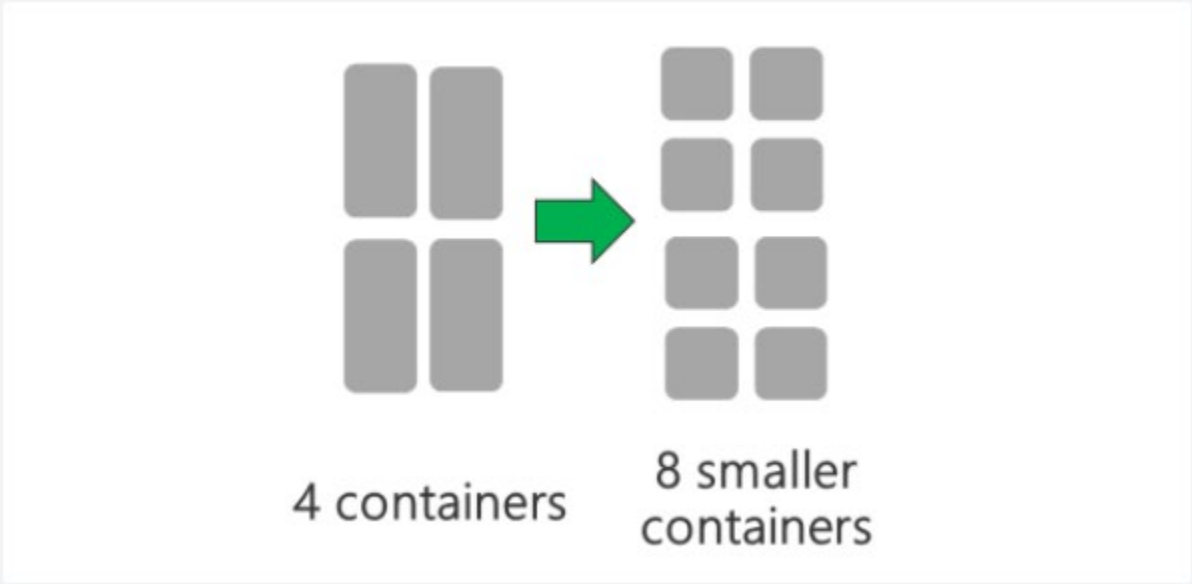
Run cluster with more nodes and/or larger sized VMs. A larger cluster will enable you to run more YARN containers as shown in the picture below.



Use VMs with more network bandwidth. The amount of network bandwidth can be a bottleneck if there is less network bandwidth than Data Lake Storage Gen2 throughput. Different VMs will have varying network bandwidth sizes. Choose a VM-type that has the largest possible network bandwidth.

YARN Layer

Use smaller YARN containers. Reduce the size of each YARN container to create more containers with the same amount of resources.

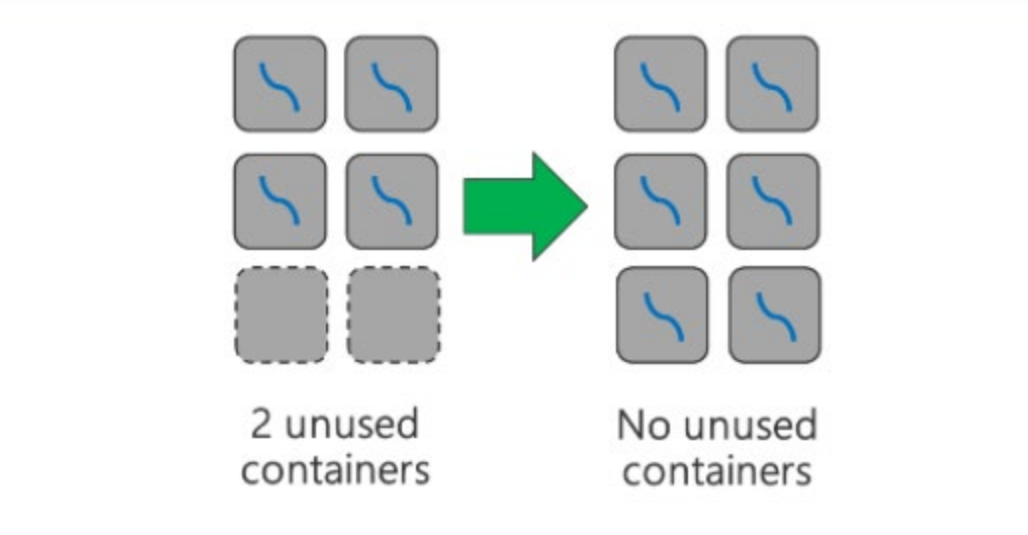


Depending on your workload, there will always be a minimum YARN container size that is needed. If you pick too small a container, your jobs will run into out-of-memory issues. Typically YARN containers should be no smaller than 1GB. It's common to see 3GB YARN containers. For some workloads, you may need larger YARN containers.

Increase cores per YARN container. Increase the number of cores allocated to each container to increase the number of parallel tasks that run in each container. This works for applications like Spark which run multiple tasks per container. For applications like Hive which run a single thread in each container, it is better to have more containers rather than more cores per container.

Workload Layer

Use all available containers. Set the number of tasks to be equal or larger than the number of available containers so that all resources are utilized.



Failed tasks are costly. If each task has a large amount of data to process, then failure of a task results in an expensive retry. Therefore, it is better to create more tasks, each of which processes a small amount of data.

https://docs.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-performance-tuning-guidance

---

Question 39: Skipped

SQL Server Integration Services (SSIS) is a platform for building complex Extract Transform and Load (ETL) solutions. SSIS is a component within SQL Server and consists of a Windows service that manages the execution of ETL workflows, along with several tools and components for developing those workflows.

SSIS is primarily a control flow engine that manages the execution of workflows. Workflows are held in packages, which can be executed [?]. (Select all that apply)

- On a schedule
(Correct)
- Only once
(Correct)
- Randomly
- On demand
(Correct)

Explanation
SQL Server Integration Services (SSIS) is a platform for building complex Extract Transform and Load (ETL) solutions. SSIS is a component within SQL Server and consists of a Windows service that manages the execution of ETL workflows, along with several tools and components for developing those workflows. It is typically used to develop data integration pipelines for on-premises data warehousing solutions. It can also be used to create data migration pipelines when migrating data between different systems.

SSIS is primarily a control flow engine that manages the execution of workflows. Workflows are held in packages, which can be executed on demand, or on a schedule (including only a single run by using the trigger now feature). Development of SSIS packages, the task workflow is referred to as the control flow of the package. A control flow can include a specific task to manage data flow operations. SSIS executes these Data Flow tasks by using a data flow engine that encapsulates the data flow in a pipeline. Each step in the Data Flow task operates in sequence on a rowset of data as it passes through the pipeline.

A SSIS solution usually consists of one or more SSIS projects, each containing one or more SSIS packages.

SSIS projects

From SQL Server 2012, a project is the unit of deployment for SSIS solutions. You can define project-level parameters to enable users to specify run-time settings, and project-level connection managers that reference data sources and destinations used in package data flows. You can then deploy projects to an SSIS Catalogue in a SQL Server instance, and configure project-level parameter values and connections as appropriate for execution environments.

SSIS packages

A project contains one or more packages, each defining a workflow of tasks to be executed. The workflow of tasks in a package is referred to as its control flow. A package control flow can include one or more Data Flow task, each of which encapsulates its own data flow pipeline. Packages can include package-level parameters so that dynamic values can be passed to the package at run time. In previous releases of SSIS, deployment was managed at the package level.

https://docs.microsoft.com/en-us/sql/integration-services/sql-server-integration-services?view=sql-server-ver15

Question 40: Skipped

To provide a better authoring experience, Azure Data Factory allows you to configure version control software for easier change tracking and collaboration. Which of the below does Azure Data Factory integrate with? (Select all that apply)

- ☐ Google Cloud Source Repositories
- ☐ Team Foundation Server
- ☐ AWS CodeCommit
- ☐ Launchpad
- ☐ Source Safe
- ☐ Git repositories
  (Correct)
- ☐ GitLab
- ☐ BitBucket
- ☐ SourceForge

Explanation

Azure Data Factory allows you to configure a Git repository with either Azure Repos or GitHub, and is a version control system that allows for easier change tracking and collaboration.

https://docs.microsoft.com/en-us/azure/data-factory/source-control

---

Question 41: Skipped

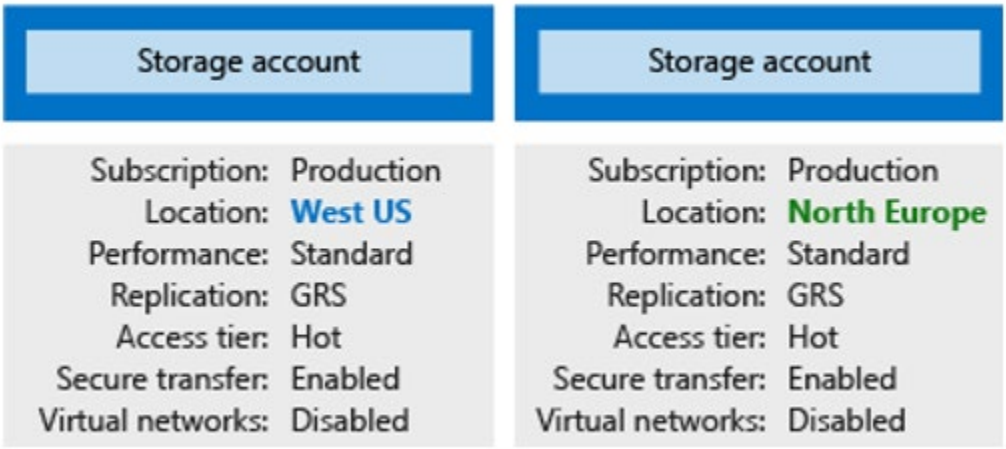
Scenario: The organization you work at has data which is specific to a country or region due to regulatory control requirements.

When considering Azure Storage Accounts, which option meets the data diversity requirement?

- Locate the organization's data it in a data centre with the strictest data regulations to ensure that regulatory requirement thresholds have been met. In this way, only one storage account will be required for managing all data, which will reduce data storage costs.
- Locate the organization's data it in a data centre in the required country or region with one storage account for each location.
  (Correct)
- Enable virtual networks for the proprietary data and not for the public data. This will require separate storage accounts for the proprietary and public data.
- None of the listed options.

Explanation
How many storage accounts do you need?

A storage account represents a collection of settings like location, replication strategy, and subscription owner. You need one storage account for every group of settings that you want to apply to your data. The following illustration shows two storage accounts that differ in one setting; that one difference is enough to require separate storage accounts.



The number of storage accounts you need is typically determined by your data diversity, cost sensitivity, and tolerance for management overhead.

Data diversity

Organizations often generate data that differs in where it is consumed, how sensitive it is, which group pays the bills, etc. Diversity along any of these vectors can lead to multiple storage accounts. Let's consider two examples:

1. Do you have data that is specific to a country or region? If so, you might want to locate it in a data centre in that country for performance or compliance reasons. You will need one storage account for each location.

2. Do you have some data that is proprietary and some for public consumption? If so, you could enable virtual networks for the proprietary data and not for the public data. This will also require separate storage accounts.

In general, increased diversity means an increased number of storage accounts.

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

---

Question 42: Skipped

Scenario: A teammate is working on solution for transferring data between a dedicated SQL Pool and a serverless Apache Spark Pool using the Azure Synapse Apache Spark Pool to Synapse SQL connector.

When could SQL Auth be used for this connection?

- When you need a token-based authentication to a dedicated SQL outside of the Synapse Analytics workspace.
  (Correct)
- None of the listed options.
- Never, it is not necessary to use SQL Auth when transferring data between a SQL or Spark Pool.
- Always, anytime you want to transfer data between the SQL and Spark Pool.

Explanation
Currently, the Azure Synapse Apache Spark Pool to Synapse SQL connector does not support a token-based authentication to a dedicated SQL pool that is outside of the workspace of Synapse Analytics. In order to establish and transfer data to a dedicated SQL pool that is outside of the workspace without Azure AD, you would have to use SQL Auth.

https://social.technet.microsoft.com/wiki/contents/articles/53259.azure-sql-three-ways-to-copy-databases-between-azure-sql-servers.aspx

---

Question 43: Skipped

True or False: In Azure Data Factory, in order to debug pipelines or activities, it is necessary to publish your workflows. Pipelines or activities which are being tested may be confined to containers to isolate them from the production environment.

- False
(Correct)
- True

Explanation

Customer requirements and expectations are changing in relation to data integration. The need among users to develop and debug their Extract Transform/Load (ETL) and Extract Load/Transform (ELT) workflows iteratively is therefore becoming more imperative.

Azure Data Factory can help you build and develop iterative debug Data Factory pipelines when you develop your data integration solution. By authoring a pipeline using the pipeline canvas, you can test your activities and pipelines by using the Debug capability.

In Azure Data Factory, there is no need to publish changes in the pipeline or activities before you want to debug. This is helpful in a scenario where you want to test the changes and see if it works as expected before you actually save and publish them.

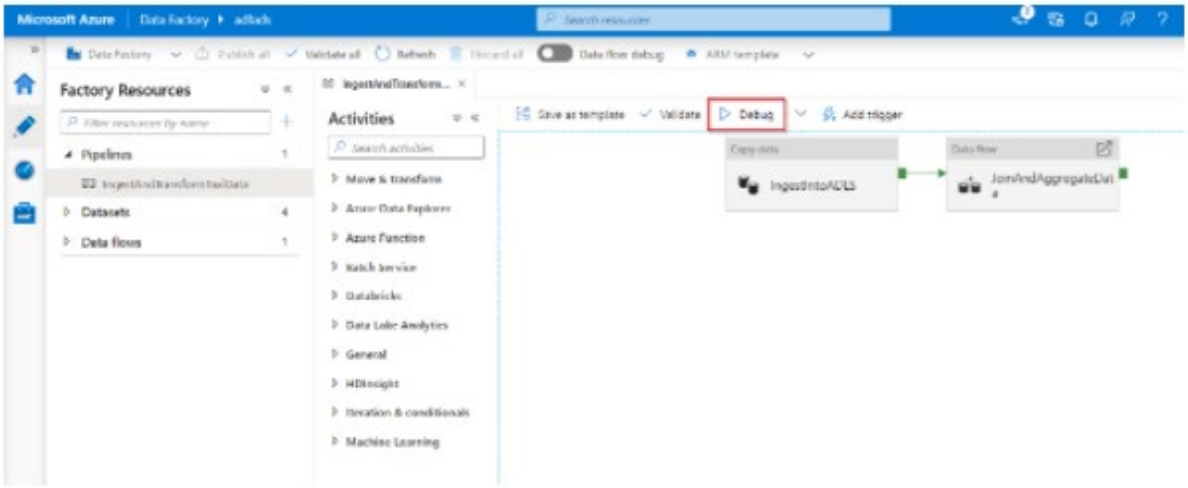
Sometimes, you don't want to debug the whole pipeline but test a part of the pipeline. A Debug run allows you to do just that. You can test the pipeline end to end or set a breakpoint. By doing so in debug mode, you can interactively see the results of each step while you build and debug your pipeline.

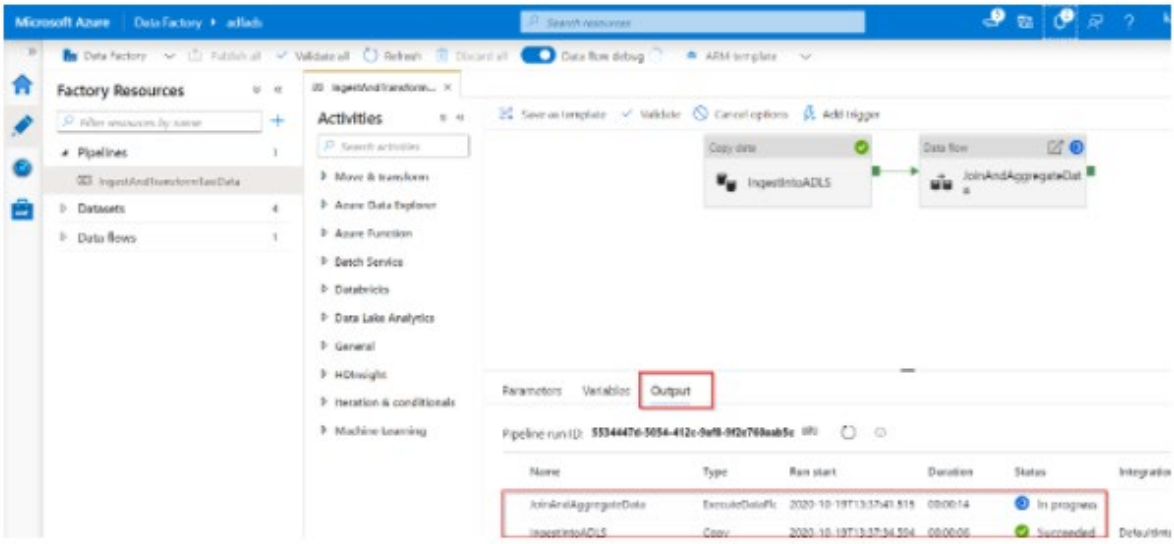
Debug and publish a pipeline:

As you create or modify a pipeline that is running, you can see the results of each activity in the Output tab of the pipeline canvas.

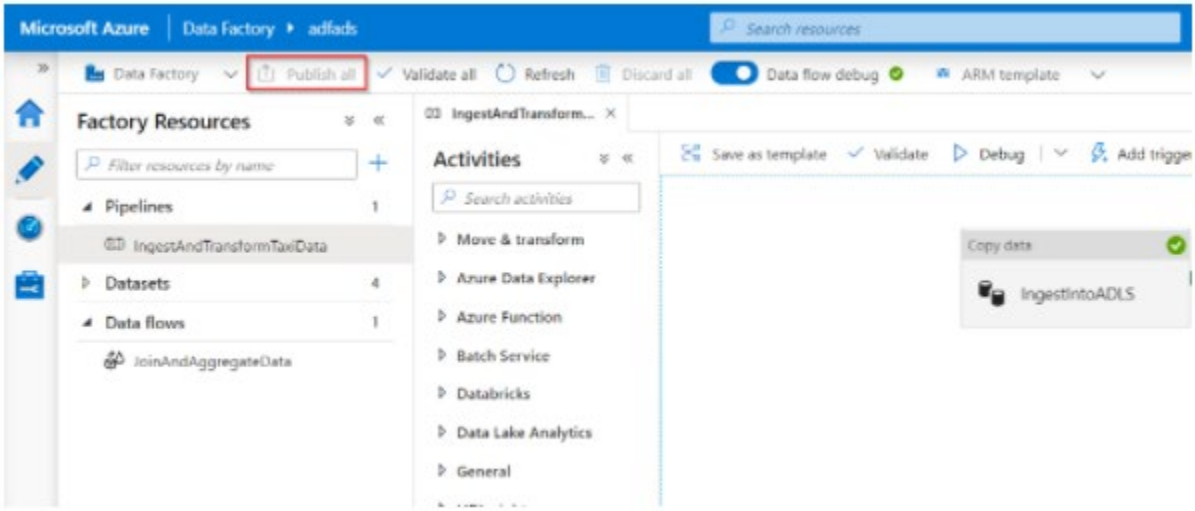
After a test run succeeds, and you are satisfied with the results, you can add more activities to the pipeline and continue debugging in an iterative manner. When you are not satisfied or like to stop the pipeline from debugging, you can cancel a test run while it is in progress. You do need to be aware that by selecting the debug slider, it will actually run the pipeline. Therefore, if the pipeline contains, for example, a copy activity, the test run will copy data from source to destination. A best practice is to use test folders in your copy activities and other activities when debugging such that when you are satisfied with the results and have debugged the pipeline, you switch to the actual folders for your normal operations.

1. To debug the pipeline, select Debug on the toolbar. You see the status of the pipeline run in the Output tab at the bottom of the window.

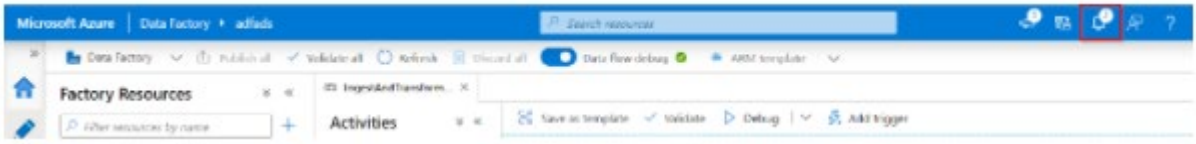




2. Once the pipeline can run successfully, in the top toolbar, select Publish all. This action publishes entities (datasets, and pipelines) you created to Data Factory.



3. Wait until you seetheSuccessfully published message. To see notification messages, click theShow Notifications on the top-right (bell button).



https://docs.microsoft.com/en-us/azure/data-factory/iterative-development-debugging

Question 44: Skipped

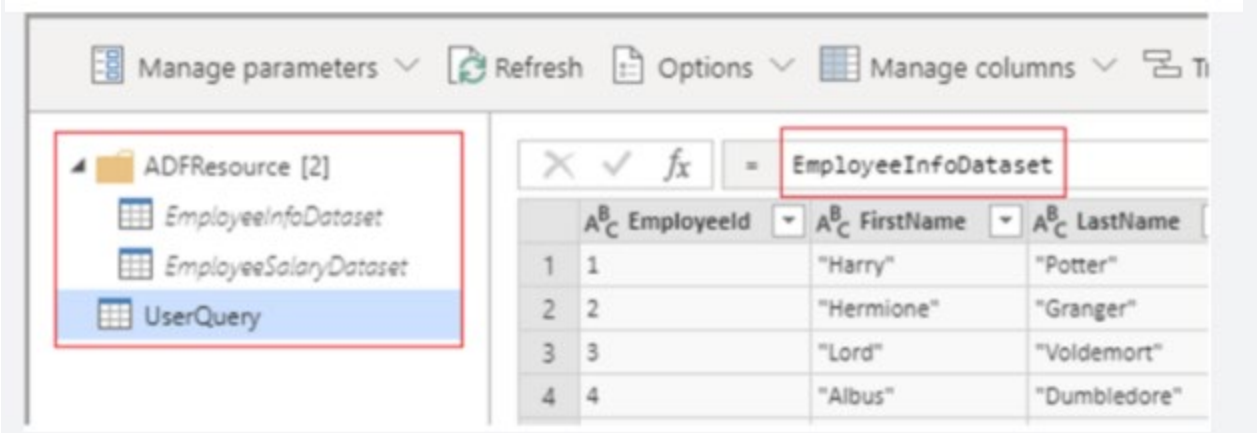
Identify the missing word(s) in the following sentence within the context of Microsoft Azure.

[?] is a data flow object that can be added to the canvas designer as an activity in an Azure Data Factory pipeline to perform code free data preparation. It enables individuals who are not conversant with the traditional data preparation technologies such as Spark or SQL Server, and languages such as Python and T-SQL to prepare data at cloud scale iteratively.

- Wrangling Data Flow
  (Correct)
- Data Stream Expression Builder
- Mapping Data Flow
- Data Expression Script Builder
- Data Expression Orchestrator
- Data Flow Expression Builder

Explanation
Wrangling Data Flow is a data flow object that can be added to the canvas designer as an activity in an Azure Data Factory pipeline to perform code free data preparation. It enables individuals who are not conversant with the traditional data preparation technologies such as Spark or SQL Server, and languages such as Python and T-SQL to prepare data at cloud scale iteratively.

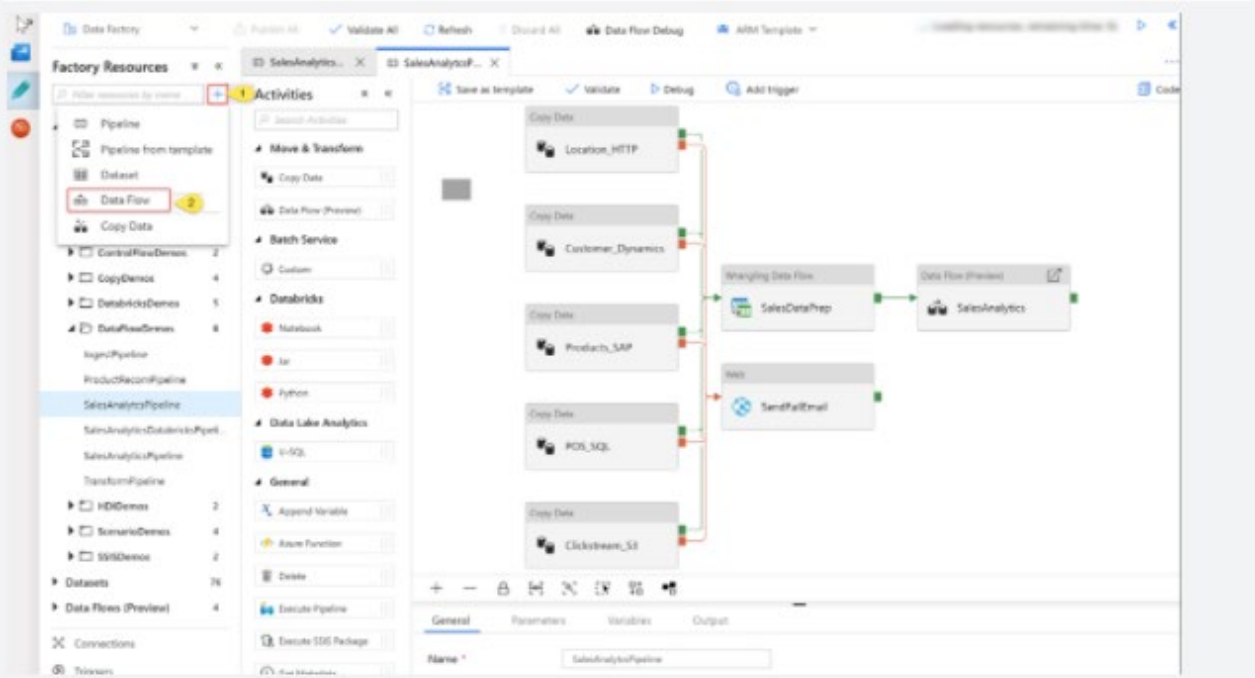
The Wrangling Data Flow uses a grid type interface for basic data preparation that is like the aesthetics of Excel, known as an Online Mashup Editor. The editor also enables more advanced users to perform more complex data preparation using formulas.



The formulas work with Power Query Online and makes Power Query M functions available for data factory users. Wrangling data flow then translates the M language generated by the Power Query Online Mashup Editor into spark code for cloud scale execution.

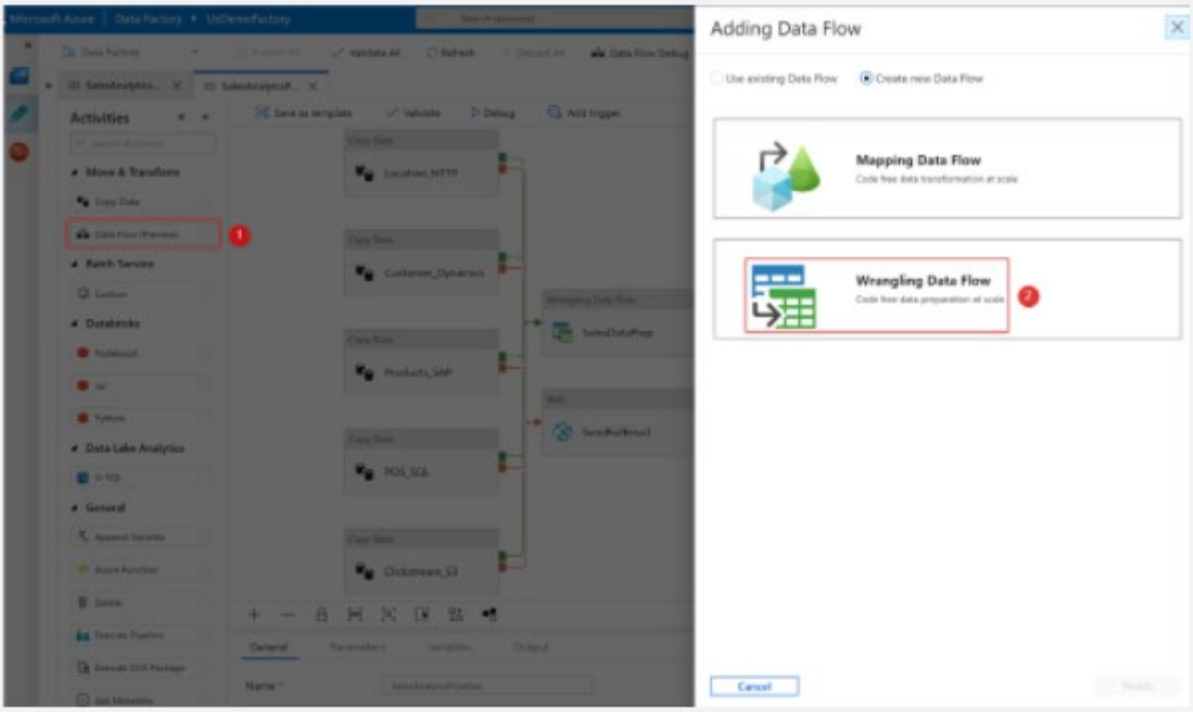
This capability enables both data engineers and citizen data integrators to interactively explore and prepare datasets. In addition, they can interactively work with the M language and preview the result before viewing it in the context of a wider pipeline.

There are two ways to create a wrangling data flow in Azure Data Factory. One way is to click the plus icon and select Data Flow in the factory resources pane.



The other method is in the activities pane of the pipeline canvas. Open the Move and Transform accordion and drag the Data flow activity onto the canvas.

In both methods, in the side pane that opens, select Create new data flow and choose Wrangling data flow. Click OK.



Add a Source dataset for your wrangling data flow, and select a sink dataset. The following data sources are supported.

Connector: Azure Blob Storage

Data format & Authentication type: CSV, Parquet | Account Key

Connector: Azure Data Lake Storage Gen1

Data format & Authentication type: CSV | Service Principal

Connector: Azure Data Lake Storage Gen2

Data format & Authentication type: CSV, Parquet | Account Key, Service Principal

Connector: Azure SQL Database

Data format & Authentication type: N/A | SQL authentication

Connector: Azure Synapse Analytics

Data format & Authentication type: N/A | SQL authentication

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

---

Question 45: Skipped

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 Databricks.

The IT team plans to create an Azure Databricks workspace that has a tiered structure. The workspace will contain the following three workloads:
• A workload for data engineers who will use Python and SQL.
• A workload for jobs that will run notebooks that use Python, Scala, and SOL.
• A workload that data scientists will use to perform ad hoc analysis in Scala and R.

The enterprise architecture team at AIM identifies the following standards for Databricks environments:
• The data engineers must share a cluster.
• The job cluster will be managed by using a request process whereby data scientists and data engineers provide packaged notebooks for deployment to the cluster.
• All the data scientists must be assigned their own cluster that terminates automatically after 120 minutes of inactivity. Currently, there are three data scientists.

Required: Create the Databricks clusters for the workloads.

Solution: The team decides to create a High Concurrency cluster for each data scientist, a High Concurrency cluster for the data engineers, and a Standard cluster for the jobs.

Does this meet the requirement?

- No
  (Correct)
- Yes

Explanation
High-concurrency clusters do not support Scala.

Standard clusters

Standard clusters are recommended for a single user. Standard clusters can run workloads developed in any language: Python, R, Scala, and SQL.

High Concurrency clusters

A High Concurrency cluster is a managed cloud resource. The key benefits of High Concurrency clusters are that they provide Apache Spark-native fine-grained sharing for maximum resource utilization and minimum query latencies.

High Concurrency clusters work only for SQL, Python, and R. The performance and security of High Concurrency clusters is provided by running user code in separate processes, which is not possible in Scala.

In addition, only High Concurrency clusters support table access control.

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

---