

Question 46: Skipped

What is the Python syntax for defining a DataFrame in Spark from an existing Parquet file in DBFS?

- ☐ `IPGeocodeDF = read.spark.parquet("dbfs:/mnt/training/ip-geocode.parquet")`
- ☒ None of the listed options
(Correct)
- ☐ `IPGeocodeDF = spark.parquet.read("dbfs:/mnt/training/ip-geocode.parquet")`
- ☐ `IPGeocodeDF = parquet.read("dbfs:/mnt/training/ip-geocode.parquet")`
- ☐ `IPGeocodeDF = spark.parquet.read("dbfs:/mnt/training/ip-geocode.parquet")`

Explanation

The correct syntax is:

```
IPGeocodeDF = spark.read.parquet("dbfs:/mnt/training/ip-geocode.parquet")
```

<https://docs.databricks.com/spark/latest/dataframes-datasets/introduction-to-dataframes-python.html>

Question 47: Skipped

Scenario: You are determining which Azure database product to use. The organization you work for needs the ability to scale up and scale down OLTP systems on demand along with Azure security and availability features.

Which of the following should be utilized?

- ☐ Azure DataNow
- ☐ Azure On-prem solution
- ☐ Azure Table Storage
- ☐ Azure Cosmos DB
- ☒ Azure SQL Database
(Correct)

Explanation

Azure SQL Database is a managed relational database service. It supports structures such as relational data and unstructured formats such as spatial and XML data. SQL Database provides online transaction processing (OLTP) that can scale on demand. You'll also find the comprehensive security and availability that you appreciate in Azure database services.

When to use SQL Database

Use SQL Database when you need to scale up and scale down OLTP systems on demand. SQL Database is a good solution when your organization wants to take advantage of Azure security and availability features. Organizations that choose SQL Database also avoid the risks of capital expenditures and of increasing operational spending on complex on-premises systems.

SQL Database can be more flexible than an on-premises SQL Server solution because you can provision and configure it in minutes. Even more, SQL Database is backed up by the Azure service-level agreement (SLA).

Key features

SQL Database delivers predictable performance for multiple resource types, service tiers, and compute sizes. Requiring almost no administration, it provides dynamic scalability with no downtime, built-in intelligent optimization, global scalability and availability, and advanced security options. These capabilities let you focus on rapid app development and on speeding up your time to market. You no longer have to devote precious time and resources to managing virtual machines and infrastructure.

<https://docs.microsoft.com/en-us/azure/azure-sql/database/sql-database-paas-overview>

Question 48: Skipped

To parallelize work, the unit of distribution is a Spark Cluster. Every Cluster has a Driver and one or more executors. Work submitted to the Cluster is split into what type of object?

- ☒ Stages
- ☐ Arrays
- ☐ Chore

- ☒ Jobs
(Correct)

Explanation

Each parallelized action is referred to as a Job. The results of each Job is returned to the Driver. Depending on the work required, multiple Jobs will be required. Each Job is broken down into Stages.



<https://www.linkedin.com/pulse/catalyst-tungsten-apache-sparks-speeding-engine-deepak-rajak?articleId=6674601890514378752>

Question 49: Skipped

Scenario: You have started at a new job within a company which has a Data Lake Storage Gen2 account. You have been tasked with moving of files from Amazon S3 to Azure Data Lake Storage.

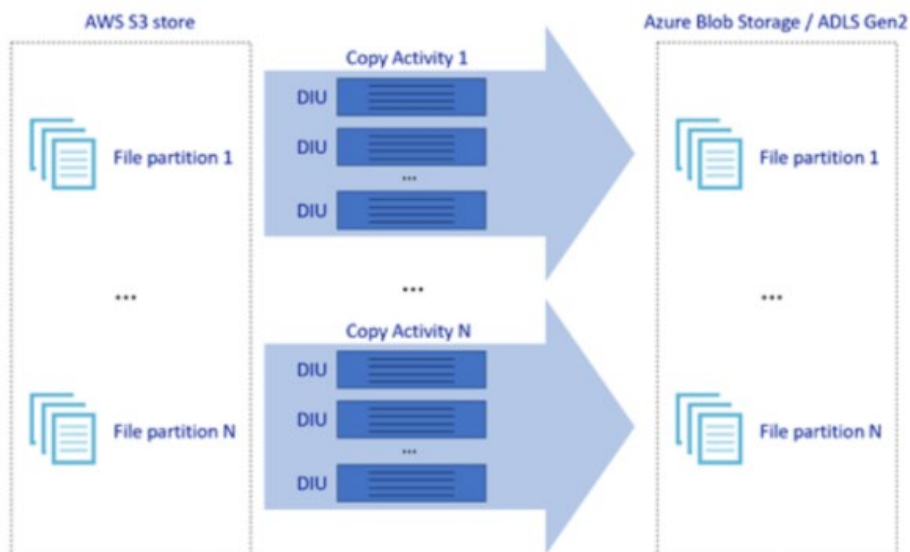
Which tool should you choose?

- ☒ Azure Data Factory
(Correct)
- ☐ Azure Data Studio
- ☐ Azure Portal

-  Azure Storage Explorer
-  Azure Data Catalog

Explanation

Azure Data Factory provides a performant, robust, and cost-effective mechanism to migrate data at scale from Amazon S3 to Azure Blob Storage or Azure Data Lake Storage Gen2.



The picture above illustrates how you can achieve great data movement speeds through different levels of parallelism:

- A single copy activity can take advantage of scalable compute resources: when using Azure Integration Runtime, you can specify **up to 256 DIUs** for each copy activity in a serverless manner; when using self-hosted Integration Runtime, you can manually scale up the machine or scale out to multiple machines (**up to 4 nodes**), and a single copy activity will partition its file set across all nodes.
- A single copy activity reads from and writes to the data store using multiple threads.
- ADF control flow can start multiple copy activities in parallel, for example using **For Each loop**.

<https://docs.microsoft.com/en-us/azure/data-factory/data-migration-guidance-s3-azure-storage>

Question 50: Skipped

What is an Azure Key Vault-backed secret scope?

- ☐ It is the Key Vault Access Key used to securely connect to the vault and retrieve secrets
- ☒ A Databricks secret scope that is backed by Azure Key Vault instead of Databricks.
(Correct)
- ☐ It is a method by which you create a secure connection to Azure Key Vault from a notebook and directly access its secrets within the Spark session
- ☐ An Azure Key Vault-backed secret scope is a private key framework managed by Microsoft.

Explanation

A secret scope is provided by Azure Databricks and can be backed by either Databricks or Azure Key Vault.

<https://docs.microsoft.com/en-us/azure/databricks/security/secrets/secret-scopes>

Question 51: Skipped

Which role works with Azure Cognitive Services, Cognitive Search, and the Bot Framework?

- ☐ A BI Engineer
- ☐ A Project Manager
- ☐ A Data Engineer
- ☐ An RPA Developer
- ☐ A Data Scientist
- ☐ A System Administrator
- ☐ A Solution Architect
- ☒ An AI Engineer
(Correct)

Explanation

Artificial intelligence (AI) engineers work with AI services such as Cognitive Services, Cognitive Search, and the Bot Framework.

AI Engineer

AI engineers work with AI services such as Cognitive Services, Cognitive Search, and Bot Framework. Cognitive Services includes Computer Vision, Text Analytics, Bing Search, and Language Understanding (LUIS).

Rather than creating models, AI engineers apply the prebuilt capabilities of Cognitive Services APIs. AI engineers embed these capabilities within a new or existing application or bot. AI engineers rely on the expertise of data engineers to store information that's generated from AI.

AI engineers add the intelligent capabilities of vision, voice, language, and knowledge to applications. To do this, they use the Cognitive Services offerings that are available out of the box.

When a Cognitive Services application reaches its capacity, AI engineers call on data scientists. Data scientists develop machine learning models and customize components for an AI engineer's application.

For example, an AI engineer might be working on a Computer Vision application that processes images. This AI engineer would ask a data engineer to provision an Azure Cosmos DB instance to store the metadata and tags that the Computer Vision application generates.

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

Question 52: Skipped

Which of the following is a good analogy for the access keys of a storage account?

- ☐ IP Address
- ☐ REST Endpoint
- ☒ Username and password
(Correct)
- ☐ Cryptographic algorithm

Explanation

Possession of an access key identifies the account and grants you access. This is very similar to login credentials like a username and password.

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

Question 53: Skipped

Which correct syntax to specify the location of a checkpoint directory when defining a Delta Lake streaming query?

- ☐

```
.writeStream.format("delta.parquet").option("checkpointLocation",  
checkpointPath) ...
```
- ☒

```
.writeStream.format("delta").option("checkpointLocation",  checkpointPath)  
...
```

(Correct)
- ☐

```
.writeStream.format("parquet").option("checkpointLocation", checkpointPath)  
...
```
- ☐

```
.writeStream.format("delta").checkpoint("location", checkpointPath) ...
```

Explanation

```
.writeStream.format("delta").option("checkpointLocation",  checkpointPath)  
...
```

 is the correct syntax to specify the checkpoint directory on a Delta Lake streaming query.

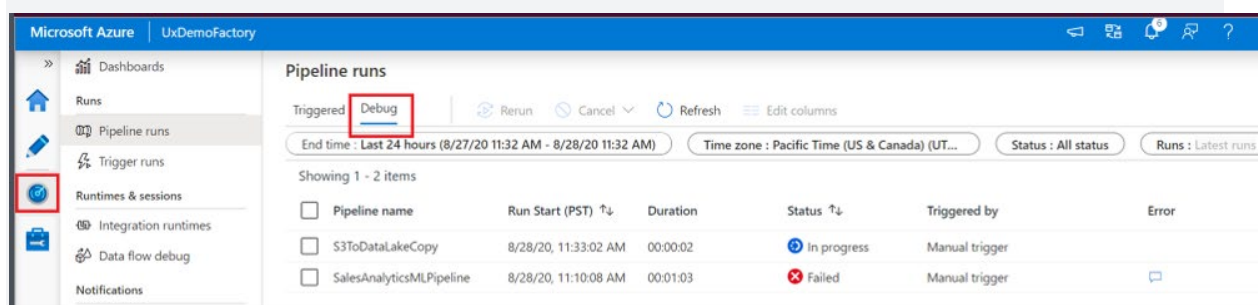
<https://docs.microsoft.com/en-us/azure/databricks/delta/delta-streaming>

Question 54: Skipped

You can monitor all of your pipeline runs natively in the Azure Data Factory user experience. The default monitoring view is list of triggered pipeline runs in the selected time period.

True or False: The list of pipeline and activity runs is auto refreshed every 60 seconds.

To view the results of a debug run, select the **Debug** tab.



True

False

(Correct)

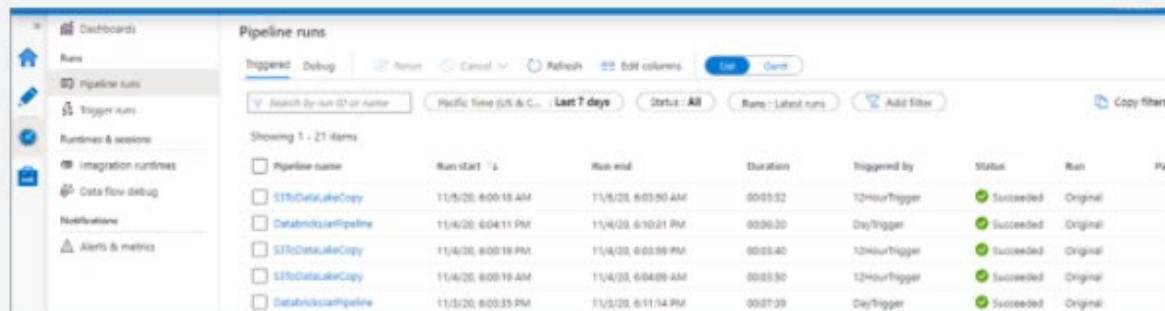
Explanation

Once you've created and published a pipeline in Azure Data Factory, you can associate it with a trigger or manually kick off an on-demand run. You can monitor all of your pipeline runs natively in the Azure Data Factory user experience. To open the monitoring experience, select the **Monitor & Manage** tile in the data factory blade of the Azure portal. If you're already in the Azure Data Factory UX, click on the **Monitor** icon on the left sidebar.

Monitor pipeline runs

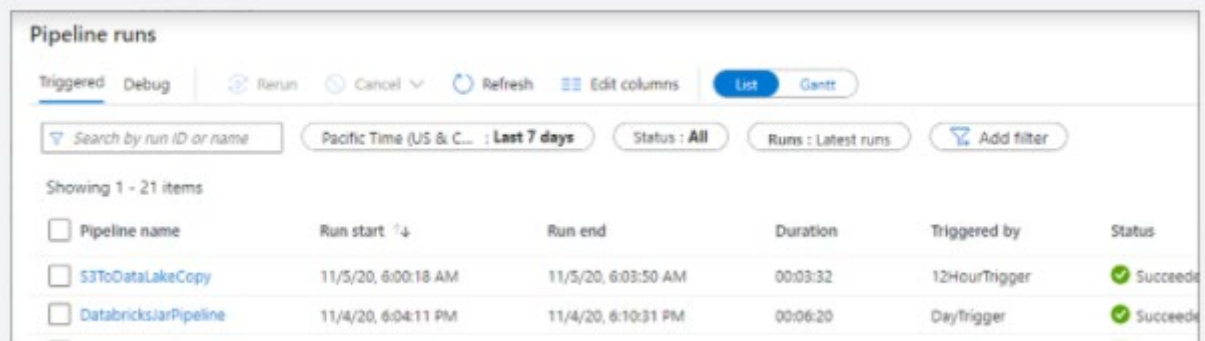
The default monitoring view is list of triggered pipeline runs in the selected time period. You can change the time range and filter by status, pipeline name, or annotation. Hover

over the specific pipeline run to get run-specific actions such as rerun and the consumption report.



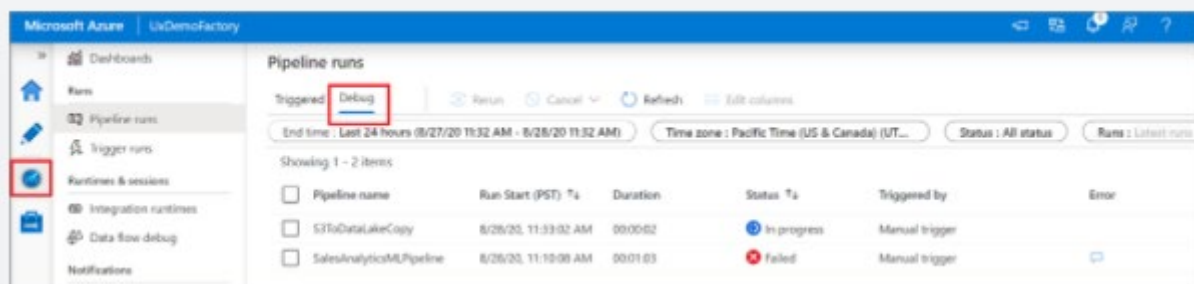
Pipeline name	Run start	Run end	Duration	Triggered by	Status	Run
S3ToDataLakeCopy	11/5/20, 6:00:18 AM	11/5/20, 6:03:50 AM	00:03:32	12HourTrigger	Succeeded	Original
DatabricksJarPipeline	11/4/20, 6:04:11 PM	11/4/20, 6:10:31 PM	00:06:20	DayTrigger	Succeeded	Original
S3ToDataLakeCopy	11/4/20, 6:00:18 PM	11/4/20, 6:03:58 PM	00:03:40	12HourTrigger	Succeeded	Original
S3ToDataLakeCopy	11/4/20, 6:00:19 AM	11/4/20, 6:04:09 AM	00:03:50	12HourTrigger	Succeeded	Original
DatabricksJarPipeline	11/5/20, 6:03:35 PM	11/5/20, 6:11:14 PM	00:07:39	DayTrigger	Succeeded	Original

You need to manually select the **Refresh** button to refresh the list of pipeline and activity runs. Autorefresh is currently not supported.



Pipeline name	Run start	Run end	Duration	Triggered by	Status
S3ToDataLakeCopy	11/5/20, 6:00:18 AM	11/5/20, 6:03:50 AM	00:03:32	12HourTrigger	Succeeded
DatabricksJarPipeline	11/4/20, 6:04:11 PM	11/4/20, 6:10:31 PM	00:06:20	DayTrigger	Succeeded

To view the results of a debug run, select the **Debug** tab.



Pipeline name	Run Start (PST)	Duration	Status	Triggered by	Error
S3ToDataLakeCopy	6/26/20, 11:33:02 AM	00:00:02	In progress	Manual trigger	
SalesAnalyticsMLPipeline	6/26/20, 11:10:08 AM	00:01:03	Failed	Manual trigger	

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

Question 55: Skipped

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

You can use a *service-level* SAS to allow access to specific resources in a storage account. You'd use this type of SAS, for example, ... [?] (Select all that apply)

- ☐ None of the listed options.
- ☐ All the listed options.
- ☒ to allow an app to retrieve a list of files in a file system.
(Correct)
- ☐ to allow the ability to create file systems.
- ☒ to allow an app to download a file.
(Correct)

Explanation

Types of shared access signatures

You can use a *service-level* SAS to allow access to specific resources in a storage account. You'd use this type of SAS, 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* SAS to allow access to anything that a service-level SAS can allow, plus additional resources and abilities. For example, you can use an account-level SAS to allow the ability to create file systems.

You'd typically use a SAS for a service where users read and write their data to your storage account. Accounts that store user data have two typical designs:

- Clients upload and download data through a front-end proxy service, which performs authentication. This front-end proxy service has the advantage of allowing validation of business rules. But, if the service must handle large amounts of data or high-volume transactions, you might find it complicated or expensive to scale this service to match demand.



- A lightweight service authenticates the client, as needed. Next, it generates a SAS. After receiving the SAS, the client can access storage account resources directly. The SAS defines the client's permissions and access interval. It reduces the need to route all data through the front-end proxy service.



<https://docs.microsoft.com/en-us/azure/storage/common/storage-sas-overview>

Question 56: Skipped

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

A transactional database must adhere to the [?] properties to ensure that the database remains consistent while processing transactions.

☐ Nuclear

☐ Atomic

☒ ACID
(Correct)

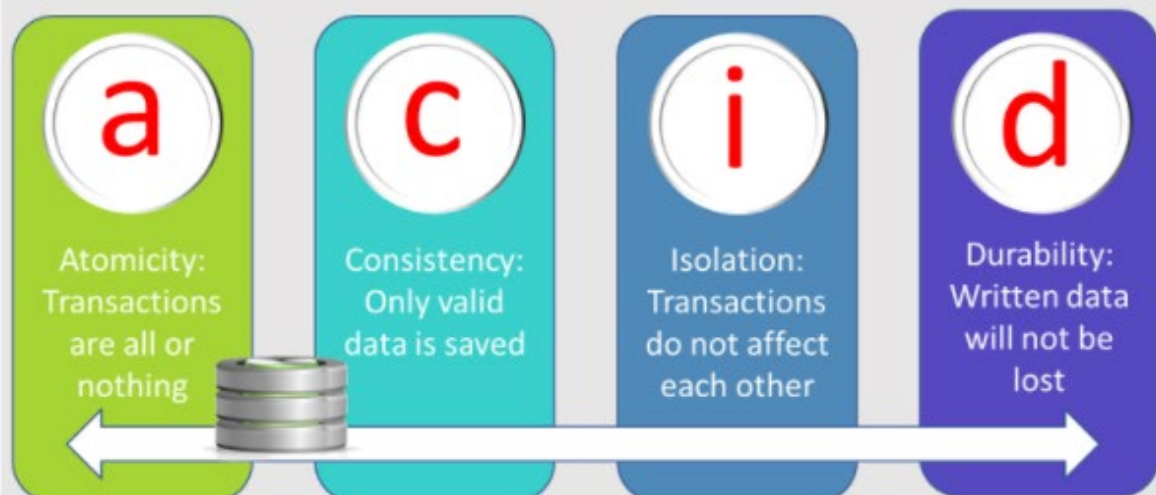
☐ Forensic

Explanation

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 57: Skipped

Scenario: You have been contracted by Wayne Enterprises, a company owned by Bruce Wayne with market value of over twenty seven million dollars. Bruce founded Wayne Enterprises shortly after he created the Wayne Foundation and he became the president and chairman of the company.

Bruce has come to you because his IT team needs advice on the proper type of storage to use for their files in an Azure Storage environment. Due to the various jurisdictions that Wayne Enterprises operates in, there are many compliance regulations which must be followed.

Required:

- A single storage account must be used to store all operations (includes all reads, writes and deletes)
- Retention policy dictates that an on-premises copy must exist for all historical operations

As the contracted expert on Azure, Bruce and the team look to you for direction. Which of the following actions will you recommend to them to meet the requirements?

- ☒ ☐ Configure the storage account to log read, write and delete operations for service type Blob
(Correct)
- ☐ ☐ Configure the storage account to log read, write and delete operations for service-type table
- ☒ ☐ Use the AzCopy tool to download log data from \$logs/blob
(Correct)
- ☐ ☐ Configure the storage account to log read, write and delete operations for service type queue
- ☐ ☐ Use the storage client to download log data from `$logs/table`

Explanation

Storage Logging logs request data in a set of blobs in a blob container named \$logs in your storage account. This container does not show up if you list all the blob containers in your account but you can see its contents if you access it directly.

Storage Analytics logs detailed information about successful and failed requests to a storage service. This information can be used to monitor individual requests and to diagnose issues with a storage service. Requests are logged on a best-effort basis. This means that most requests will result in a log record, but the completeness and timeliness of Storage Analytics logs are not guaranteed.

Storage Analytics logging is not enabled by default for your storage account. You can enable it in the [Azure portal](#) or by using PowerShell, or Azure CLI. For step-by-step guidance, see [Enable and manage Azure Storage Analytics logs \(classic\)](#).

You can also enable Storage Analytics logs programmatically via the REST API or the client library. Use the [Get Blob Service Properties](#), [Get Queue Service Properties](#), and [Get Table Service Properties](#) operations to enable Storage Analytics for each service. To see an example that enables Storage Analytics logs by using .NET, see [Enable logs](#)

Log entries are created only if there are requests made against the service endpoint. For example, if a storage account has activity in its Blob endpoint but not in its Table or Queue endpoints, only logs pertaining to the Blob service will be created.

<https://docs.microsoft.com/en-us/rest/api/storageservices/enabling-storage-logging-and-accessing-log-data>

To view and analyze your log data, you should download the blobs that contain the log data you are interested in to a local machine. Many storage-browsing tools enable you to download blobs from your storage account; you can also use the Azure Storage team provided command-line Azure Copy Tool (AzCopy) to download your log data.

AzCopy is a command-line utility that you can use to copy blobs or files to or from a storage account. This article helps you download AzCopy, connect to your storage account, and then transfer files.

<https://docs.microsoft.com/en-us/azure/storage/common/storage-use-azcopy-v10>



<https://www.youtube.com/watch?v=GJYAgi5eYYE>

Question 58: Skipped

How can all notebooks in Synapse studio be saved?

- ☐ Notebooks are synced to the Synapse Studio cloud automatically upon changes being made to a file.
- ☐ Select the Publish button on the notebook command bar.
- ☒ Select the Publish all button on the workspace command bar.
(Correct)
- ☐ Using CTRL + S

Explanation

To save all notebooks in your workspace, select the Publish all button on the workspace command bar.

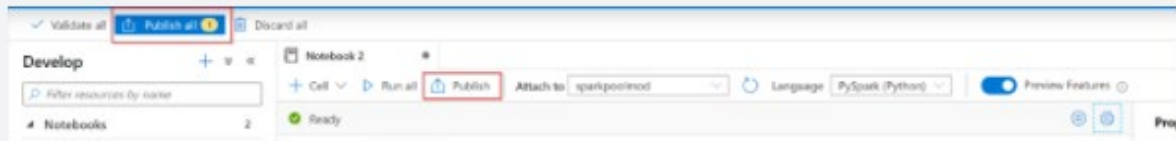
It is possible to save a single or all notebooks that you've created with Azure Synapse Studio notebooks.

You have the possibility to save a single notebook or all notebooks in your workspace.

To save changes you made to a single notebook, select the **Publish** button on the notebook command bar.



To save all notebooks in your workspace, select the **Publish all** button on the workspace command bar.



In the notebook properties, you can configure whether to include the cell output when

A screenshot of the 'Properties' dialog for a notebook. The 'General' tab is selected. It contains a message: 'Choose a name for your Notebook. This name can be updated at any time until it is published.' Below this are fields for 'Name' (containing 'Notebook 2') and 'Description'. The 'Type' is '.ipynb notebook' and the 'Size' is '1,386 bytes'. The 'Notebook settings' section has a checkbox 'Include cell output when saving' which is checked and highlighted with a red box. At the bottom, there is a 'Session' section with a link 'Configure session'.

saving.

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

Question 59: Skipped

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

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

Internally, Azure Kubernetes Service (AKS) is used to ... [?]

- ☐ specify the types and sizes of the virtual machines.
- ☐ provide the fastest virtualized network infrastructure in the cloud.
- ☒ run the Azure Databricks control-plane and data-planes via containers running on the latest generation of Azure hardware.
(Correct)
- ☐ auto-scale as needed based on your usage and the setting used when configuring the cluster.
- ☐ pulls data from a specified data source.

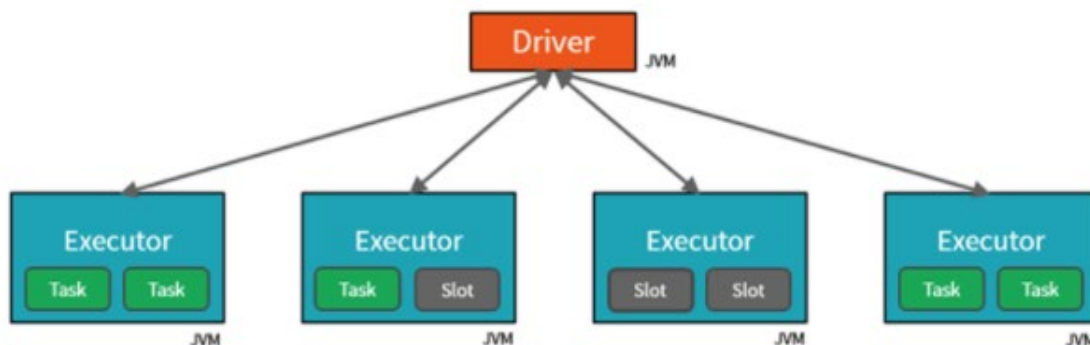
Explanation

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

High-level overview

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

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



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

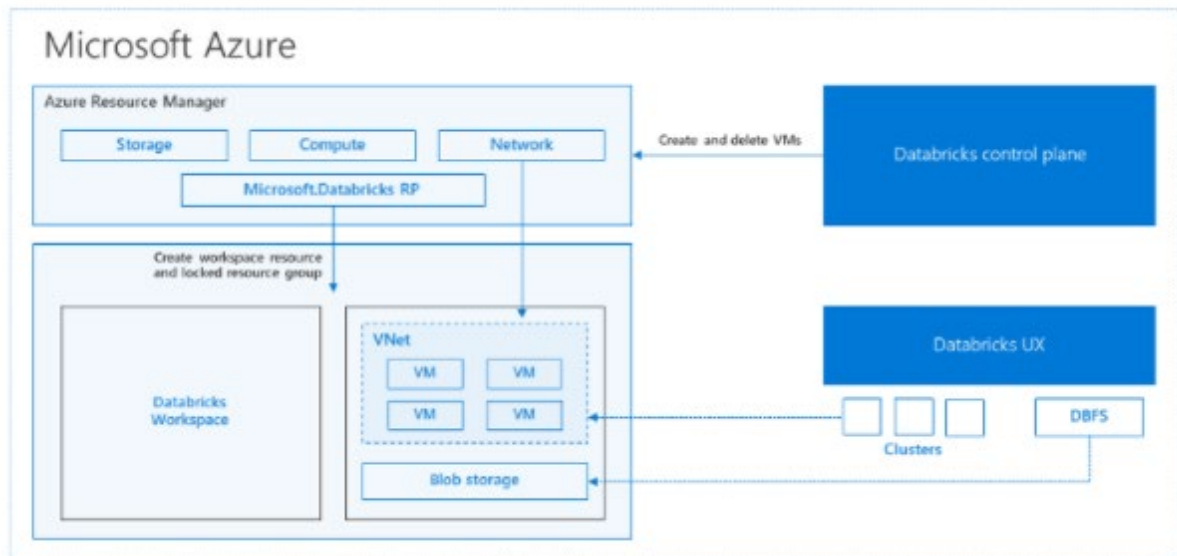
Under the covers

Now let's take a deeper look under the covers. When you create an Azure Databricks service, a "Databricks appliance" is deployed as an Azure resource in your subscription. At the time of cluster creation, you specify the types and sizes of the virtual machines (VMs) to use for both the Driver and Worker nodes, but Azure Databricks manages all other aspects of the cluster.

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

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

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



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

Question 60: Skipped

Scenario: You are working as a consultant at Avengers Security. At the moment, you are consulting with Tony, the lead of the IT team and the topic of discussion is access provisioning for an Azure Data Lake Storage Gen2 account.

Quentin Beck is a team member who has contributor access to the storage account, as well as the application ID access key. One of Quentin's tasks on his to-do list is to use PolyBase to load data into Azure SQL data warehouse.

Required: Configure PolyBase to connect the data warehouse to the storage account.

Tony has listed out a few items that he thinks Quentin should create to perform the task, but is not sure if they are correct and is not sure of the order of operations needed to complete the requirement successfully.

- a. A database encryption key
- b. An asymmetric key
- c. An external data source
- d. An external file format
- e. A database scoped credential

Since you are an Azure SME, he looks to you for advice to identify the correct items to create and for you to arrange them in the correct order.

Which of the following identifies the correct items needed in the correct order to fulfill the requirement?

- ☐ c → e → a → d
- ☐ a → d → c
- ☐ c → d → a → e
- ☒ e → c → d
(Correct)
- ☐ a → d → c → b → e

Explanation

Step 1: A database scoped credential

To access your Data Lake Storage account, you will need to create a Database Master Key to encrypt your credential secret used in the next step. You then create a database scoped credential.

Step 2: An external data source

Create the external data source. Use the CREATE EXTERNAL DATA SOURCE command to store the location of the data. Provide the credential created in the previous step.

Step 3: An external file format

Configure data format: To import the data from Data Lake Storage, you need to specify the External File Format. This object defines how the files are written in Data Lake Storage.

Load data from Azure Data Lake Storage into dedicated SQL pools in Azure Synapse Analytics

Create the target table

Connect to your dedicated SQL pool and create the target table you will load to. In this example, we are creating a product dimension table.

```
SQL
-- A: Create the target table
-- DimProduct
CREATE TABLE [dbo].[DimProduct]
(
    [ProductKey] [int] NOT NULL,
    [ProductLabel] [nvarchar](255) NULL,
    [ProductName] [nvarchar](500) NULL
)
WITH
(
    DISTRIBUTION = HASH([ProductKey]),
    CLUSTERED COLUMNSTORE INDEX
    --HEAP
);
```

Create the COPY statement

Connect to your SQL dedicated pool and run the COPY statement. For a complete list of examples, visit the following documentation: [Securely load data using dedicated SQL pools](#).

```
SQL
-- B: Create and execute the COPY statement

COPY INTO [dbo].[DimProduct]

--The column list allows you map, omit, or reorder input file columns to target table columns.
```

```

--You can also specify the default value when there is a NULL value in the file.

--When the column list is not specified, columns will be mapped based on source and target ordinality
(
    ProductKey default -1 1,
    ProductLabel default 'myStringDefaultWhenNull' 2,
    ProductName default 'myStringDefaultWhenNull' 3
)

--The storage account location where your data is staged
FROM 'https://storageaccount.blob.core.windows.net/container/directory/'
WITH
(
    --CREDENTIAL: Specifies the authentication method and credential access your storage account
    CREDENTIAL = (IDENTITY = '', SECRET = ''),
    --FILE_TYPE: Specifies the file type in your storage account location
    FILE_TYPE = 'CSV',
    --FIELD_TERMINATOR: Marks the end of each field (column) in a delimited text (CSV) file
    FIELDTERMINATOR = '|',
    --ROWTERMINATOR: Marks the end of a record in the file
    ROWTERMINATOR = '0x0A',
    --FIELDQUOTE: Specifies the delimiter for data of type string in a delimited text (CSV) file
    FIELDQUOTE = '',
    ENCODING = 'UTF8',
    DATEFORMAT = 'ymd',
    --MAXERRORS: Maximum number of reject rows allowed in the load before the COPY operation is canceled
    MAXERRORS = 10,
    --ERRORFILE: Specifies the directory where the rejected rows and the corresponding error reason should be written
    ERRORFILE = '/errorsfolder',
) OPTION (LABEL = 'COPY: ADLS tutorial');

```

Optimize columnstore compression

By default, tables are defined as a clustered columnstore index. After a load completes, some of the data rows might not be compressed into the columnstore. There's a variety of reasons why this can happen. To learn more, see [manage columnstore indexes](#).

To optimize query performance and columnstore compression after a load, rebuild the table to force the columnstore index to compress all the rows.

SQL

```
ALTER INDEX ALL ON [dbo].[DimProduct] REBUILD;
```

Optimize statistics

It is best to create single-column statistics immediately after a load. There are some choices for statistics. For example, if you create single-column statistics on every column it might take a long time to rebuild all the statistics. If you know certain columns are not going to be in query predicates, you can skip creating statistics on those columns.

If you decide to create single-column statistics on every column of every table, you can use the stored procedure code sample `prc_sqldw_create_stats` in the [statistics](#) article.

<https://docs.microsoft.com/en-us/azure/synapse-analytics/sql-data-warehouse/sql-data-warehouse-load-from-azure-data-lake-store#create-the-target-table>

Question 61: Skipped

Once Azure Synapse Link is configured on Cosmos DB, what is the first step to perform to use Azure Synapse Analytics serverless SQL pools to query the Azure Cosmos DB data?

- ☐ None of the listed options
- ☐ Use the `OPENROWSET` function
- ☒ `CREATE` database
(Correct)
- ☐ Use a `SELECT` clause

Explanation

Before being able to issue any queries using Azure Synapse Analytics serverless SQL pools, you first must create a database.

Question 62: Skipped

Scenario: You are working as a consultant at Avengers Security and advising the IT team on the design of a hybrid solution to synchronize data and on-premises Microsoft SQL Server database to Azure SQL Database.

Required: An assessment of databases must be done in order to determine whether or not data will move without compatibility issues.

The Avengers IT team has many different tools at their disposal and it is your responsibility to advise them on which tool to use. Which of the following is the best for the application?

- ☐ Microsoft Assessment and Planning Toolkit
- ☐ SQL Vulnerability Assessment (VA)
- ☐ SQL Server Migration Assistant (SSMA)
- ☒ Data Migration Assistant (DMA)
(Correct)

Explanation

The Data Migration Assistant (DMA) helps you upgrade to a modern data platform by detecting compatibility issues that can impact database functionality in your new version of SQL Server or Azure SQL Database. DMA recommends performance and reliability improvements for your target environment and allows you to move your schema, data, and uncontained objects from your source server to your target server.

Data Migration Assistant is a client-side tool that you can install on a Windows-compatible workstation or server. It has two major functions in the migration of the social database to the Azure SQL Database platform in this module.

- First, it assesses your existing database and identifies any incompatibilities between that database and Azure SQL Database.
- It then generates a report of the things you need to fix before you can migrate.

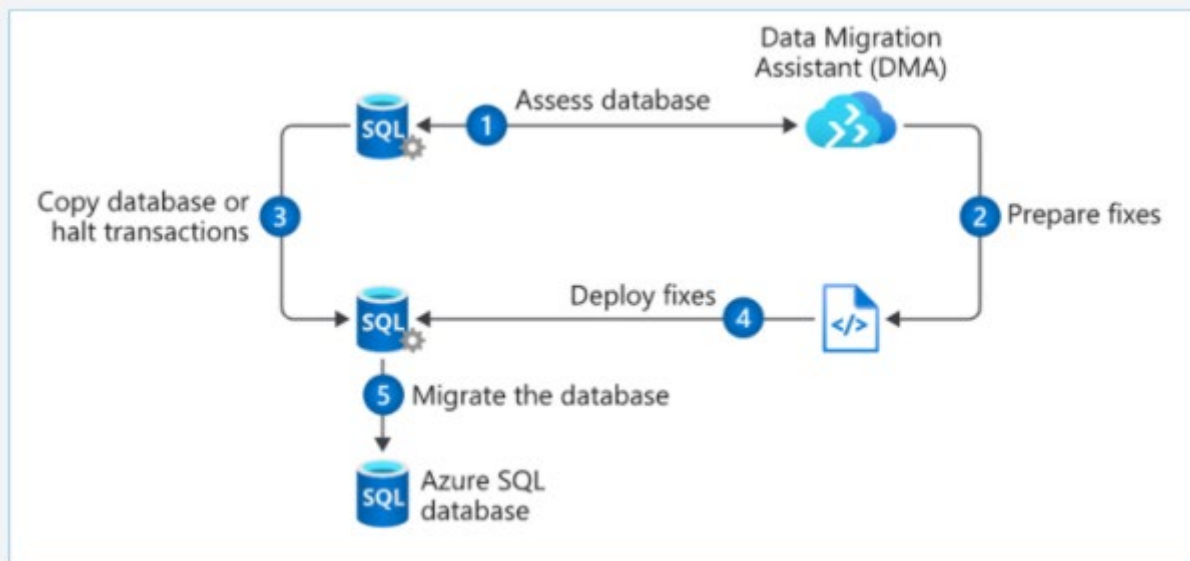
As you make changes, you can rerun Data Migration Assistant to generate an updated report of changes that you need to make. This capability helps you to not only track your progress, but also catch any new issues that might have been introduced during your coding phase.

Migration process overview

Migrating your company's social media database is a multi-step process. The workflow begins with a *pre-migration* phase, in which you determine which databases need to be migrated. You also look for any compatibility issues between your existing database and Azure SQL Database.

After you resolve any incompatibility issues, you're ready for the *migration* phase. First, you migrate the schema to the Azure SQL Database Service. Then, you're ready to migrate the data itself by using Azure Database Migration Service.

The last step in your workflow is the *post-migration* phase. During this phase, you do any required testing. Then you update apps, reports, and other tools that will need to use the new database for their data.



Pre-migration

The pre-migration phase begins with *discovery*, or taking inventory of your existing databases and the tools and apps that rely on them. For this simple exercise, we're concerned with only a single social database. In practice, it can be a much more complex step.

You need to identify everything that uses your existing database. Apps, SQL Server Report Services reports, Power BI reports, and export jobs written in PowerShell are all examples of things to note so you can update them, after the migration, to point to the new Azure SQL Database.

The second step in the pre-migration phase is the *assessment*. During the assessment, you examine the database for any incompatibilities between it and the Azure SQL Database platform. Because this can be a difficult task to perform manually, Microsoft has provided Data Migration Assistant. You can use Data Migration Assistant to automatically examine your source database for any compatibility issues with Azure SQL Database.

Data Migration Assistant provides a report that you can use as a guide to update your database. As you make changes, you can rerun Data Migration Assistant to track your progress and to uncover any new issues that might arise as you make changes. The assessment phase is covered in steps 1 and 2 of the migration workflow previously illustrated.

The final stage in the pre-migration is *convert*. In the convert phase, you make any changes for compatibility that Data Migration Assistant has recommended. Then, you create the SQL scripts for deploying to the Azure SQL Database. Data Migration

Assistant can be of help to you here as well. It generates all of the SQL scripts needed to deploy your schema to the target Azure SQL Database.

Migration

The migration phase involves migrating two elements: *schema* and *data*. In the convert phase of pre-migration, the Data Migration Assistant tool generated all of the code. Data Migration Assistant can run these scripts for you. Or, you can save these scripts, and run them on your own by using a tool such as SQL Server Management Studio, Azure Data Studio, or the `sqlcmd` utility. The schema migration can be found in step 4 of the migration workflow.

After your database schema has been migrated, you're ready to migrate your data (steps 3 and 5 in the workflow). For this step, you'll use Azure Database Migration Service to move your data up to the Azure SQL Database Service.

Database Migration Service can be run in two modes, online and offline. When it's running in online mode, there are two additional steps. The first is *sync*, in which any changes made to the data in the source system after the migration are brought into the target database. The other is *cutover*, in which the source database is taken offline, and the new Azure SQL Database becomes available.

Post-migration

The post-migration phase is a process that consists of several steps. First, you need to remediate any applications, updating any affected by the database changes. For example, you might need to update the connection strings to point to the new Azure SQL Database.

In addition, make sure there's thorough and complete testing. Validation testing will ensure that your application did not break because of changes at the database level. Construct tests to return data from both the source and target. Compare the data to ensure that queries are returning from the Azure SQL Database as they would with the original source database. Next, create performance tests that will:

- Validate that your application returns data in the times required by your organization.
- Enable you to do further optimizations, if necessary.

The post-migration phase is critical because it ensures that your data is both accurate and complete. In addition, it alerts you to any performance issues that might arise with the workload in the new environment.

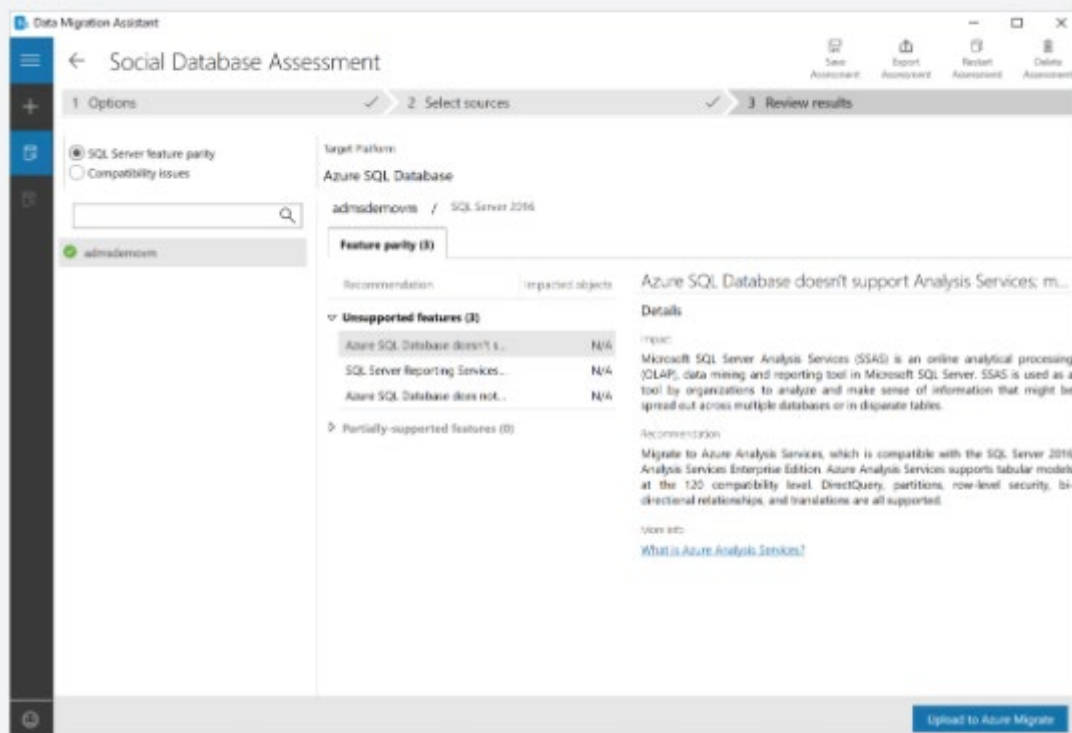
Data migration tools in Azure

The core of data migration in Azure is the Azure Database Migration Service. You can use this service to move bulk amounts of data in a timely way. As part of Database

Migration Service, Microsoft provides Data Migration Assistant. Just as its name implies, Data Migration Assistant assists the service by preparing the target database.

Data Migration Assistant

Data Migration Assistant is a client-side tool that you can install on a Windows-compatible workstation or server. It has two major functions in the migration of the social database to the Azure SQL Database platform in this module.



First, it assesses your existing database and identifies any incompatibilities between that database and Azure SQL Database. It then generates a report of the things you need to fix before you can migrate. As you make changes, you can rerun Data Migration Assistant to generate an updated report of changes that you need to make. This capability helps you to not only track your progress, but also catch any new issues that might have been introduced during your coding phase.

After Data Migration Assistant completes the assessment and you've made any changes, you need to migrate the database schema to Azure SQL Database. Data Migration Assistant can help with this as well. It generates the required SQL, and then gives you the option of running the code, or saving it so you can run it yourself later.

Using Data Migration Assistant is not a requirement to use Azure Database Migration Service. You have the option of coding your new database in the Azure SQL Database service manually without trying to convert an existing database.

As an example, let's say you're creating a staging database in Azure SQL Database that will later feed data into Azure Synapse Analytics. The staging database will be sourced from multiple systems, but it will migrate only small portions of the source data. In this situation, you might be better off manually crafting the new database directly on the Azure SQL Database service rather than trying to automate the job.

Azure Database Migration Service

After you've migrated your database schema by using Data Migration Assistant, or created a target database manually, you're ready to move your data. To do that, you'll use Azure Database Migration Service.

Azure Database Migration Service is a fully-managed Azure service that provides automated, seamless data migrations from multiple sources into the Azure data platforms.

The screenshot shows the Azure Database Migration Service interface. At the top, there's a search bar and a toolbar with buttons: '+ New Migration Project', 'Delete service', 'Refresh', 'Start Service', and 'Stop Service'. A green banner at the top states: 'Great job! Your database migration service was successfully created. You can create your first migration project now.'

Below the banner, there's an 'Essentials' section with a table of service details:

Resource group (change)	Status
edmacarg	Online
Network & Ip Address	Location
edmac-vnet/subnets/default 10.0.0.4	westus
Subscription name (change)	Subscription ID
SQL DB Content	< subscription id >
SKU	Service/UI Version
Basic 1 vCore	3.4.4038.1/3.4.4038.1

Below the 'Essentials' section, there's a 'Migration Projects' section with a table header:

NAME	SOURCE	TARGET	CREATED
No database migration projects to display			

At the bottom, there's another green banner with the same message: 'Great job! Your database migration service was successfully created. You can create your first migration project now.' and a blue button labeled 'New migration project'.

Database Migration Service runs on the Azure platform, as opposed to being a client application like Data Migration Assistant. It's capable of moving large amounts of data quickly and is not dependent upon installation of a client application. Database Migration Service can operate in two modes, offline and online.

In offline mode, no more changes can be made to your source database. Data is migrated, and then your applications can begin using the new Azure SQL Database.

In online mode, your source database can remain in use while the bulk of the data is migrated. At the end of the migration, you'll take the source system offline momentarily

while any final changes to the source are synced to the new Azure SQL Database. At this point, your applications can cut over to use the SQL database.

<https://docs.microsoft.com/en-us/sql/dma/dma-overview>

Question 63: Skipped

A pipeline in Azure Data Factory represents a logical grouping of activities where the activities together perform a certain task.

Which of the following are valid dependency conditions? (Select four)

- ☒ Completed
(Correct)
- ☐ Pending
- ☐ Working
- ☐ Queue
- ☒ Succeeded
(Correct)
- ☒ Skipped
(Correct)
- ☒ Failed
(Correct)
- ☐ Running

Explanation

Azure Data Factory

In order to work with data factory pipelines, it is imperative to understand what a pipeline in Azure Data Factory is.

A pipeline in Azure Data Factory represents a logical grouping of activities where the activities together perform a certain task.

An example of a combination of activities in one pipeline can be, ingesting and cleaning log data in combination with a mapping data flow that analyzes the log data that has been cleaned.

A pipeline enables you to manage the separate individual activities as a set, which would otherwise be managed individually. It enables you to deploy and schedule the activities efficiently, through the use of a single pipeline, versus managing each activity independently.

Activities in a pipeline are referred to as actions that you perform on your data. An activity can take zero or more input datasets and produce one or more output datasets.

An example of an action can be the use of a copy activity, where you copy data from an Azure SQL Database to an Azure DataLake Storage Gen2. To build on this example, you can use a data flow activity or an Azure Databricks Notebook activity for processing and transforming the data that was copied to your Azure Data Lake Storage Gen2 account, in order to have the data ready for business intelligence reporting solutions like in Azure Synapse Analytics.

Since there are many activities that are possible in a pipeline in Azure Data Factory, we have grouped the activities in three categories:

- *Data movement activities*: the Copy Activity in Data Factory copies data from a source data store to a sink data store.
- *Data transformation activities*: Azure Data Factory supports transformation activities such as Data Flow, Azure Function, Spark, and others that can be added to pipelines either individually or chained with another activity.
- *Control activities*: Examples of control flow activities are 'get metadata', 'For Each', and 'Execute Pipeline'.

Activities can depend on each other. What we mean, is that the activity dependency defines how subsequent activities depend on previous activities. The dependency itself can be based on a condition of whether to continue in the execution of previous defined activities in order to complete a task. An activity that depends on one or more previous activities, can have different dependency conditions.

The four dependency conditions are:

- Succeeded
- Failed
- Skipped
- Completed

For example, if a pipeline has an Activity A, followed by an Activity B and Activity B has as a dependency condition on Activity A 'Succeeded', then Activity B will only run if Activity A has the status of succeeded.

If you have multiple activities in a pipeline and subsequent activities are not dependent on previous activities, the activities may run in parallel.

<https://datasavvy.me/2021/02/18/azure-data-factory-activity-failures-and-pipeline-outcomes/>

Question 64: Skipped

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

Data classification: Structured

Operations: Read-only, complex analytical queries across multiple databases

Latency & throughput: Some latency in the results is expected based on the complex nature of the queries.

Transactional support: Not required

- ☐ Azure Route Table
- ☐ Azure Cosmos DB
- ☐ Azure Queue Storage
- ☐ Azure Blob Storage
- ☒ Azure SQL Database
(Correct)

Explanation

Recommended service: Azure SQL Database

Business data will most likely be queried by business analysts, who are more likely to know SQL than any other query language. Azure SQL Database could be used as the solution by itself, but pairing it with Azure Analysis Services enables data analysts to create a semantic model over the data in SQL Database. The data analysts can then share it with business users, so that they only need to connect to the model from any business intelligence (BI) tool to immediately explore the data and gain insights.

Why not other Azure services?

Azure Synapse supports OLAP solutions and SQL queries. But your business analysts will need to perform cross-database queries, which Azure Synapse does not support.

Azure Analysis Services could be used in addition to Azure SQL Database. But your business analysts are more well-versed in SQL than in working with Power BI. So they'd like a database that supports SQL queries, which Azure Analysis Services does not. In addition, the financial data you're storing in your business data set is relational and multidimensional in nature. Azure Analysis Services supports tabular data stored on the service itself, but not multidimensional data. To analyze multidimensional data with Azure Analysis Services, you can use a direct query to the SQL Database.

Azure Stream Analytics is a great way to analyze data and transform it into actionable insights, but its focus is on real-time data that is streaming in. In this scenario, the business analysts are looking at historical data only.

<https://docs.microsoft.com/en-us/azure/azure-sql/database/sql-database-paas-overview>

Question 65: Skipped

Scenario: You are working in an Azure Databricks workspace and you want to filter by a `productType` column where the value is equal to `book`.

Which command meets the requirement by specifying a column value in a DataFrame's filter?

- ☐ `df.filter("productType == 'book'")`
- ☒ `df.filter(col("productType") == "book")`
(Correct)
- ☐ `df.col("productType").filter("book")`

- ☐ `df.filter("productType = 'book'")`

Explanation

The `df.filter(col("productType") == "book")` approach is the correct way to apply the filter, by using the Column Class.

<https://docs.databricks.com/spark/latest/dataframes-datasets/introduction-to-dataframes-python.html>

Question 66: Skipped

Scenario: You are working on a new project and creating storage accounts and blob containers for your application.

Which of the below describes a good strategy for doing this?

- ☒ Create Azure Storage accounts before deploying your app. Create containers in your application as needed.
(Correct)
- ☐ Create Azure Storage accounts in your application as needed. Create the containers before deploying the application.
- ☐ Create both your Azure Storage accounts and containers before deploying your application.
- ☐ None of the listed options.

- ☐ All the listed options.

Explanation

Creating an Azure Storage account is an administrative activity and can be done prior to deploying an application. Container creation is lightweight and is often driven by run-time data which makes it a good activity to do in your application.

<https://docs.microsoft.com/en-us/azure/storage/common/storage-account-create?tabs=azure-portal>

Question 67: Skipped

Scenario: You work in an organization where much of the transformation logic is currently held in existing SSIS packages that have been created on SQL Server. Since your boss is not familiar with Azure as well as you are, he tells you he has heard that Azure has the ability to lift and shift SSIS package so to execute them within Azure Data Factory to leverage existing work. He asks you *"What do we need to setup in order to do this?"*

Which of the below is the correct response?

- ☐ None of the listed options.
- ☒ In order to do this you must set up an Azure-SSIS integration runtime.
(Correct)
- ☐ In order to do this you must set up a Self-hosted solution and then upload the data.
- ☐ Your boss is mistaken, Azure does not have the ability to lift and shift SSIS package so to execute them within Azure Data Factory, it must be converted to AZ format and then ingested via Azure Storage.
- ☐

In order to do this you must set up an Azure Stored procedure to execute the lift and shift.

Explanation

You may work in an organization where much of the transformation logic is currently held in existing SSIS packages that have been created on SQL Server. You have the ability to lift and shift SSIS package so you can execute them within Azure Data Factory, so you can make use in existing work. In order to do this you must set up an Azure-SSIS integration runtime.

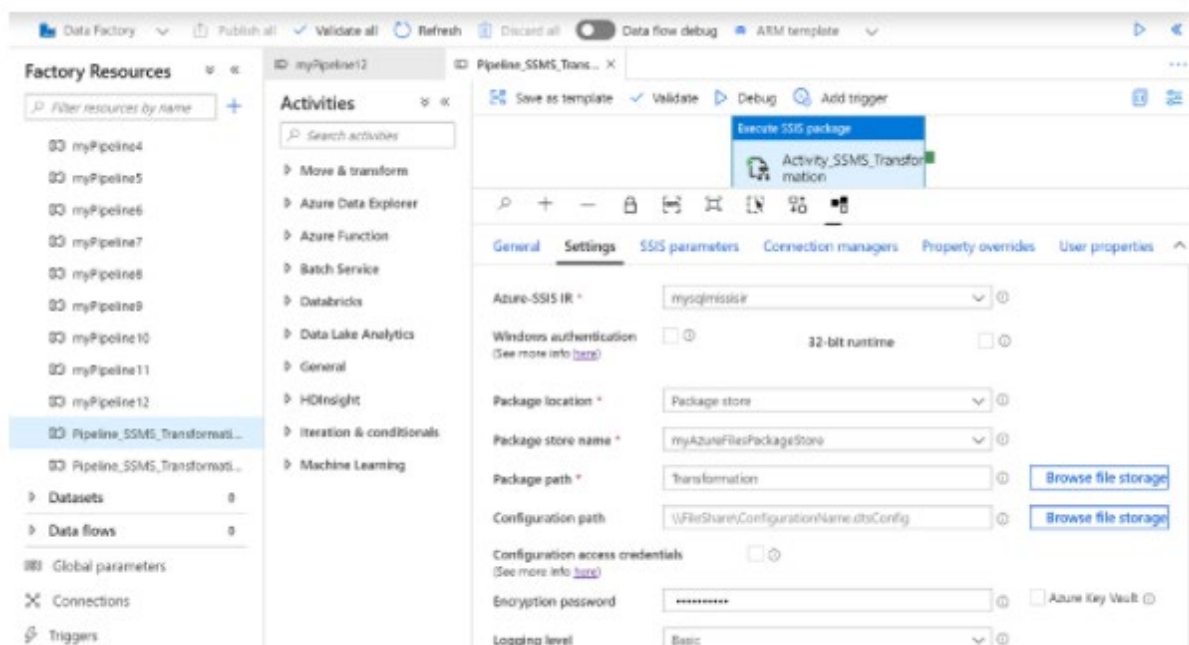
Azure-SSIS integration runtime

In order to make use of the Azure-SSIS integration runtime, it is assumed that there is SSIS Catalog (SSISDD) deployed on a SQL Server SSIS instance. With that prerequisite met, the Azure-SSIS integration runtime is capable of:

- Lift and shift existing SSIS workloads

During the provisioning of the Azure-SSIS integration runtime, you specify the following options:

- The node size (including the number of cores) and the number of nodes in the cluster.
- The existing instance of Azure SQL Database to host the SSIS Catalog Database (SSISDB), and the service tier for the database.
- The maximum parallel executions per node.



With the Azure-SSIS integration runtime enabled, you are able to manage, monitor and schedule SSIS packages using tools such as SQL Server Management Studio (SSMS) or SQL Server Data Tools (SSDT).

<https://docs.microsoft.com/en-us/azure/data-factory/azure-ssis-integration-runtime-package-store>

Question 68: Skipped

When is it possible to add or remove datasets if created with Azure Data Share?

- ☐ It is not possible to add or remove datasets if created with Azure Data Share.
- ☐ It is only possible to remove or add datasets before it's sent within Azure Data Share.
- ☐ None of the listed options.
- ☒ It is possible to add or remove datasets within Azure Data Share after it has been created.

(Correct)

Explanation

It is possible to add or remove datasets after it has been created in Azure Data Share.

<https://docs.microsoft.com/en-us/azure/data-factory/lab-data-flow-data-share>

Question 69: Skipped

Azure Cosmos DB is a globally distributed, multimodel database. Which of the following can be used to deploy it?

- ☒ Cassandra API
(Correct)
- ☒ Gremlin API
(Correct)
- ☐ T-SQL API
- ☒ Table API
(Correct)
- ☒ SQL API
(Correct)
- ☐ ABS API
- ☐ ADLS API
- ☐ U-SQL API
- ☒ MongoDB API
(Correct)

Explanation

Azure Cosmos DB is a globally distributed, multimodel database. You can deploy it by using several API models:

- SQL API
- MongoDB API
- Cassandra API
- Gremlin API
- Table API

Because of the multimodel architecture of Azure Cosmos DB, you benefit from each model's inherent capabilities. For example, you can use MongoDB for semistructured data, Cassandra for wide columns, or Gremlin for graph databases. When you move your data from SQL, MongoDB, or Cassandra to Azure Cosmos DB, applications that are built using the SQL, MongoDB, or Cassandra APIs will continue to operate.

<https://docs.microsoft.com/en-us/azure/cosmos-db/faq>

Question 70: Skipped

In Data Factory, an Activity defines the action to be performed. A Linked Service defines a target data store or a compute service. An Integration Runtime (IR) provides the bridge between the Activity and Linked Services.

In order to make use of the Azure-SSIS Integration Runtime, it is assumed that there is SSIS Catalog (SSISDB) deployed on a SQL Server SSIS instance. With that prerequisite met, the Azure-SSIS Integration Runtime is capable of lifting and shifting existing SSIS workloads

During the provisioning of the Azure-SSIS Integration Runtime, which are the options that must be specified? (Select all that apply)

- ☐ All the listed options
- ☒ Node size
(Correct)
- ☐ IP address(es) of the nodes
- ☐ VM regions
- ☒ Database (SSISDB) along with the service tier for the database
(Correct)
- ☐ Private Link parameters
- ☒ Maximum parallel executions per node
(Correct)
- ☒ Existing instance of Azure SQL Database to host the SSIS Catalog
(Correct)

Explanation
Integration Runtime

In Data Factory, an Activity defines the action to be performed. A Linked Service defines a target data store or a compute service. An Integration Runtime (IR) provides the bridge between the Activity and Linked Services.

Azure-SSIS Integration Runtime - To lift and shift existing SSIS workload, you can create an Azure-SSIS IR to natively execute SSIS packages. Selecting the right location for your Azure-SSIS IR is essential to achieve high performance in your extract-transform-load (ETL) workflows.

- The location of your Azure-SSIS IR does not need to be the same as the location of your data factory, but it should be the same as the location of your own Azure SQL Database or Azure SQL Database managed instance server where SSISDB is to be hosted. This way, your Azure-SSIS Integration Runtime can easily access SSISDB without incurring excessive traffics between different locations.
- If you do not have an existing Azure SQL Database or Azure SQL Database managed instance server to host SSISDB, but you have on-premises data sources/destinations, you should create a new Azure SQL Database or Azure SQL Database managed instance server in the same location of a virtual network connected to your on-premises network. This way, you can create your Azure-SSIS IR using the new Azure SQL Database or Azure SQL Database managed instance server and joining that virtual network, all in the same location, effectively minimizing data movements across different locations.
- If the location of your existing Azure SQL Database or Azure SQL Database managed instance server where SSISDB is hosted is not the same as the location of a virtual network connected to your on-premises network, first create your Azure-SSIS IR using an existing Azure SQL Database or Azure SQL Database managed instance server and joining another virtual network in the same location, and then configure a virtual network to virtual network connection between different locations.

In order to make use of the Azure-SSIS Integration Runtime, it is assumed that there is SSIS Catalog (SSISDB) deployed on a SQL Server SSIS instance. With that prerequisite met, the Azure-SSIS Integration Runtime is capable of lifting and shifting existing SSIS workload. **During the provisioning of the Azure-SSIS Integration Runtime, you specify the following options:**

- **The node size (including the number of cores) and the number of nodes in the cluster.**
- **The existing instance of Azure SQL Database to host the SSIS Catalog Database (SSISDB), and the service tier for the database.**
- **The maximum parallel executions per node.**

With the Azure-SSIS Integration Runtime enabled, you are able to manage, monitor, and schedule SSIS packages using tools such as SQL Server Management Studio (SSMS) or SQL Server Data Tools (SSDT).

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

Question 71: Skipped

True or False: Azure Storage encrypts all data that's written to it. It is not necessary to enable encryption within your subscription.

- ☒ True
(Correct)
- ☐ False

Explanation

Azure Storage Data security

Azure Storage encrypts all data that's written to it. Azure Storage also provides you with fine-grained control over who has access to your data. You'll secure the data by using keys or shared access signatures.

Azure Resource Manager provides a permissions model that uses role-based access control (RBAC).

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

Question 72: Skipped

Data engineers use Azure Stream Analytics to process streaming data and respond to data anomalies in real time. You can use Stream Analytics for Internet of Things (IoT) monitoring, web logs, remote patient monitoring, and point of sale (POS) systems.

Stream Analytics can route job output to which of the following storage systems? (Select all that apply)

☐ Azure SQL Datawarehouse

☐ Azure Storage Explorer

☒ Azure SQL Database
(Correct)

☒ Azure Data Lake Storage
(Correct)

☒ Azure Cosmos DB
(Correct)

☐ Azure Table Storage

☒ Azure Blob Storage
(Correct)

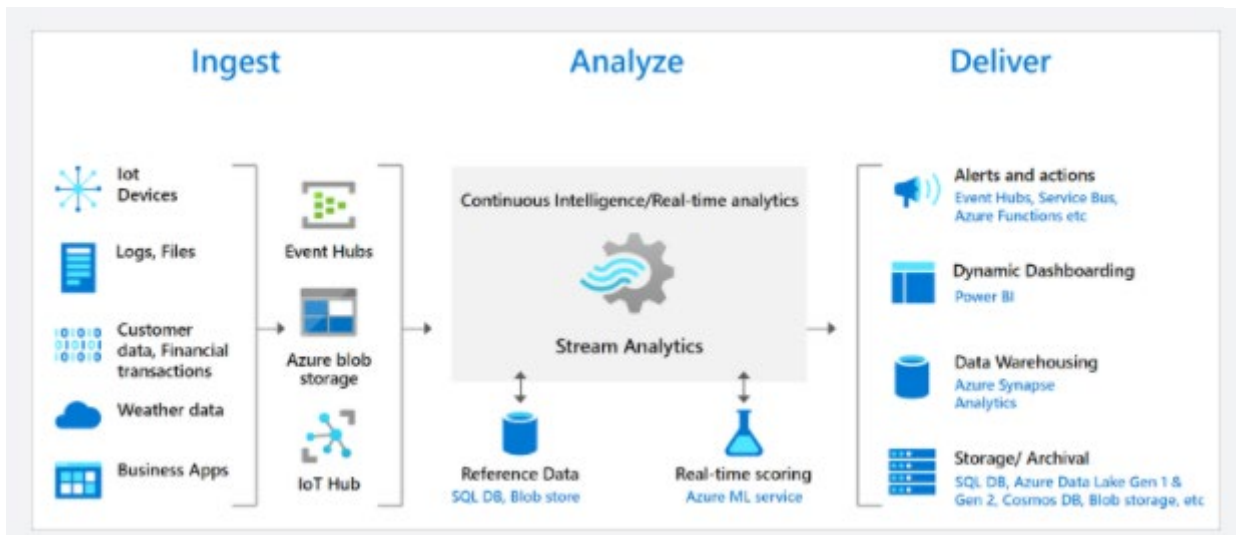
Explanation

Applications, sensors, monitoring devices, and gateways broadcast continuous event data known as *data streams*. Streaming data is high volume and has a lighter payload than nonstreaming systems.

Data engineers use Azure Stream Analytics to process streaming data and respond to data anomalies in real time. You can use Stream Analytics for Internet of Things (IoT) monitoring, web logs, remote patient monitoring, and point of sale (POS) systems.

Data processing

To process streaming data, set up Stream Analytics jobs with input and output pipelines. Inputs are provided by Event Hubs, IoT Hubs, or Azure Storage. Stream Analytics can route job output to many storage systems. These systems include Azure Blob, Azure SQL Database, Azure Data Lake Storage, and Azure Cosmos DB.



After storing the data, run batch analytics in Azure HDInsight. Or send the output to a service like Event Hubs for consumption. Or use the Power BI streaming API to send the output to Power BI for real-time visualization.

Queries

To define job transformations, use a simple, declarative Stream Analytics query language. The language should let you use simple SQL constructs to write complex temporal queries and analytics.

The Stream Analytics query language is consistent with the SQL language. If you're familiar with the SQL language, you can start creating jobs.

Data security

Stream Analytics handles security at the transport layer between the device and Azure IoT Hub. Streaming data is generally discarded after the windowing operations finish. Event Hubs uses a shared key to secure the data transfer. If you want to store the data, your storage device will provide security.

<https://docs.microsoft.com/en-us/azure/stream-analytics/stream-analytics-introduction>

Question 73: Skipped

What is a step in flattening a nested schema?

- ☐ **LOAD** CSV file
- ☐ **CREATE** parquet file
- ☐ **CREATE** Delta Lake table

Explode Arrays (Correct)

-  data

Explanation

Explode Arrays is a third step in flattening nested schema's. It is necessary to transform the array in the data frame into a new dataframe where the column that you want to select is defined.

A DataFrame creates a data structure and it's one of the core data structures in Spark. In Spark, it is seen as a distributed collection of data that is organized into columns that have names.

Some use cases for transforming complex data types are as follows:

- Complex data types are increasingly common and represent a challenge for data engineers as analyzing nested schema and arrays tend to include time-consuming and complex SQL queries.
- It can be difficult to rename or cast the nested columns data type.
- Performance issues arise when working with deeply nested objects.
- Data Engineers need to understand how to efficiently process complex data types and make them easily accessible to everyone.

Synapse Spark can be used to read and transform objects into a flat structure through data frames. Synapse SQL serverless can be used to query such objects directly and return those results as a regular table. With Synapse Spark, it's easy to transform nested structures into columns and array elements into multiple rows.

In the overview below, the steps show the techniques involved to deal with complex data types:



- Step 1: Define a function for flattening We define a function to flatten the nested schema.
- Step 2: Flatten nested schema Use the function to flatten the nested schema of the data frame (df) into a new data frame.
- Step 3: Explode Arrays Transform the array in the data frame into a new dataframe where you also define the column that you want to select.
- Step 4: Flatten child nested Schema Use the function you create to flatten the nested schema of the data frame into a new data frame.

https://medium.com/@saikrishna_55717/flattening-nested-data-json-xml-using-apache-spark-75fa4c8ea2a7

Question 74: Skipped

Azure Data Lake Storage Gen2 plays a fundamental role in a wide range of big data architectures. There are stages for processing big data solutions that are common to all architectures.

Which are they? (Select four)

☐ Model and serve
(Correct)

☐ Ingestion
(Correct)

☐ Streamed

- ☐ Store
(Correct)
- ☐ Relational
- ☐ Prep and train
(Correct)
- ☐ Clusters

Explanation

Azure Data Lake Storage Gen2 plays a fundamental role in a wide range of big data architectures. These architectures can involve the creation of:

- A modern data warehouse.
- Advanced analytics against big data.
- A real-time analytical solution.

There are four stages for processing big data solutions that are common to all architectures:

- **Ingestion** - The ingestion phase identifies the technology and processes that are used to acquire the source data. This data can come from files, logs, and other types of unstructured data that must be put into the Data Lake Store. The technology that is used will vary depending on the frequency that the data is transferred. For example, for batch movement of data, Azure Data Factory may be the most appropriate technology to use. For real-time ingestion of data, Apache Kafka for HDInsight or Stream Analytics may be an appropriate technology to use.

- **Store** - The store phase identifies where the ingested data should be placed. In this case, we're using Azure Data Lake Storage Gen2.

- **Prep and train** - The prep and train phase identifies the technologies that are used to perform data preparation and model training and scoring for data science solutions. The common technologies that are used in this phase are Azure Databricks, Azure HDInsight or Azure Machine Learning Services.

- **Model and serve** - Finally, the model and serve phase involves the technologies that will present the data to users. These can include visualization tools such as Power BI, or other data stores such as Azure Synapse Analytics, Azure Cosmos DB, Azure SQL Database, or Azure Analysis Services. Often, a combination of these technologies will be used depending on the business requirements.

Question 75: Skipped

Which Dynamic Management View enables the view the active connections against a dedicated SQL pool?

- ☐ `sys.dm_pdw_nodes_os_connection_counters`
- ☐ `sys.dm_pdw_dms_workers`
- ☐ `DBCC PDW_SHOWEXECUTIONPLAN`
- ☐ `sys.dm_pdw_exec_sessions`
- ☐ `sys.dm_pdw_nodes_exec_connection`
- ☐ `sys.dm_pdw_dms_workers`
- ☐ `sys.dm_pdw_request_steps`

- ☒ `sys.dm_pdw_exec_requests`
(Correct)

Explanation

`sys.dm_pdw_exec_requests` enables you to view the active connections against a dedicated SQL pool

<https://docs.microsoft.com/en-us/sql/relational-databases/system-dynamic-management-views/sys-dm-pdw-exec-requests-transact-sql?view=aps-pdw-2016-au7>

Question 76: Skipped

Which is one of the possible ways to optimize a Spark Job?

- ☐ Remove all nodes
- ☐ Remove the Spark Pool
- ☐ Use the local cache option
- ☒ Use bucketing
(Correct)
- ☐ None of the listed options

Explanation

The way bucketed tables are optimized is because it's because the metadata about how it was bucketed and sorted are stored.

Once you have checked the monitor tab within the Azure Synapse Studio environment, and feel that you could improve the performance of the run, you have several things to take in mind:

- Choose the data abstraction

- Use the optimal data format
- Use the cache option
- Check the memory efficiency
- Use Bucketing
- Optimize Joins and Shuffles if appropriate
- Optimize Job Execution

In order to optimize the Apache Spark Jobs in Azure Synapse Analytics, you need to take into account the cluster configuration for the workload you're running on that cluster. You might run into challenges where memory pressure (if not configured well, like not choosing the right size of executors), long running operations and tasks that might result in Cartesian operations. If you want to speed up the jobs, you'd have to configure the appropriate caching for that task, as well as checking joins and shuffles in relation to data skew. Therefore, it is so imperative that you monitor and review Spark Job executions that are long running or resource-consuming.

Some recommendations in order for you to optimize the Spark Job might include the following:

Choosing the data abstraction

Some of the earlier Spark versions use RDDs to abstract the data. Spark 1.3 and 1.6 introduced the use of DataFrames and Datasets. The following relative merits might help you to optimize in relation to your data abstraction:

DataFrames Using DataFrames would be a great place to start. DataFrames provide query optimization through Catalyst. It also includes a whole-stage code generation with direct memory access. One thing to take in mind is that when you want to have the best-developer-friendly experience it might be better to use Datasets, since there are no compile-time checks or domain object programming.

On that note, let's look into Datasets: *Datasets are good in complex ETL pipelines optimization where the performance impact is acceptable. Just be cautious when using Datasets in aggregations, since it might impact the performance. However, it will provide query optimization through Catalyst and is developer-friendly by providing object programming and compile-time checks. Datasets do add serialization/deserialization overhead and has a high GC overhead.

Looking at RDDs we would advise as follows: It is not necessary to use RDDs unless you want or need to build a new custom RDD. However, there is no query optimization through Catalyst as well as no whole-stage code generation and would still have a high GC overhead. The only way to use RDDs is that it needs SPark 1.x legacy APIs.

When looking at your data format, spark provides many. Formats that you can use are csv, json, xml, parquet etc. It can also be extended by other formats with external data sources. A tip that might be useful is using parquet with snappy compression (which also happen to be the default in Spar 2.x.) Why Parquet? It stores data in a columnar format, is compressed and highly optimized in Spark, as well as, splittable in order to decompress.

When it comes to the caching, there is a native built in Spark caching mechanism. It can be used through different methods like: `.persist()`, `.cache()`, and `CACHE` `TABL` E. When using small datasets, it might be effective. In ETL pipelines where caching of intermediate results is necessary this might come in handy too. Just take in mind that is you need to do partitioning, the spark native caching mechanism might have some downsides. The reason for that is that a cached table won't keep the partitioning data.

It is also imperative to understand how to use the memory efficiently. What you have to understand is that Spark operates by placing data in memory. Therefore, managing memory resources is an aspect of optimizing Spark jobs executions. The way to manage it, might be to check smaller data partitions and checking data size, types and distributions when you formulate a partitioning strategy. Another way to optimize is to consider Kryo data serialization: [Kryo data serialization](#), versus the default Java serialization. Always bear in mind though, to keep monitoring and tuning the Spark configuration settings.

Another thing to look at might be bucketing.

Use bucketing

Bucketing is almost the same as data partitioning. The way it differs is that a bucket holds a set of column values instead of one. It might work well when you partition on large (millions or more) values like product identifiers. A bucket is determined by hashing the bucket key of a row. The way bucketed tables are optimized is because it's because the metadata about how it was bucketed and sorted are stored.

Some advanced bucketing features are:

- Query optimization based on bucketing meta-information.
- Optimized aggregations.
- Optimized joins.

However, bucketing doesn't exclude partitioning. They go hand in had. You can use partitioning and bucketing at the same time.

Optimize joins and shuffles

Sometimes, when you have a slower performance on join or shuffle jobs, it can be caused by data skew. What is data skew? It's asymmetry in your job data. An example might be that a job only takes 20 sec regularly, however running the same job where data is joined and shuffled can take up hours. In order to fix that data skew, you can salt the entire key, or use an isolated salt for only some subset of keys. Another option to look into might be the introduction of a bucket column and pre-aggregate in buckets first. However, there's more to causing slow joins, since it might be the join type. Spark uses the SortMerge join type. This type of join is best suited for large data sets, but is otherwise computationally expensive because it must first sort the left and right sides of data before merging them. Therefore, a Broadcast join might be better suited for smaller data sets, or where one side of the join is much smaller than the other side.

You can change the join type in your configuration by setting `spark.sql.autoBroadcastJoinThreshold`, or you can set a join hint using the DataFrame APIs (`dataframe.join(broadcast(df2))`).

```
Scala
// Option 1
spark.conf.set("spark.sql.autoBroadcastJoinThreshold", 1*1024*1024*1024)

// Option 2
val df1 = spark.table("FactTableA")
val df2 = spark.table("dimMP")
df1.join(broadcast(df2), Seq("PK")).
createOrReplaceTempView("V_JOIN")

sql("SELECT col1, col2 FROM V_JOIN")
```

If you did decide to use bucketed tables, you will have a third join type, the Merge join. A correctly pre-partitioned and pre-sorted dataset will skip the expensive sort phase from a SortMerge join. Another thing to take in mind is that the order of the different type of joins does matter, especially in complex queries. Therefore, it's advised to start with the most selective joins. In addition, try to move joins that increase the number of rows after aggregations when possible.

Looking at the sizing of executors in order to increase performance in your spark job, you could consider the Java garbage Collection Overhead (GC) overhead.

- Factors to reduce executor size:
 - Reduce heap size below 32 GB to keep GC overhead < 10%.
 - Reduce the number of cores to keep GC overhead < 10%.

- Factors to increase executor size:

- Reduce communication overhead between executors.
 - Reduce the number of open connections between executors (N2) on larger clusters (>100 executors).
 - Increase heap size to accommodate for memory-intensive tasks.
 - Optional: Reduce per-executor memory overhead.
 - Optional: Increase utilization and concurrency by oversubscribing CPU.

As a general rule of thumb when selecting the executor size:

- Start with 30 GB per executor and distribute available machine cores.
- Increase the number of executor cores for larger clusters (> 100 executors).
- Modify size based both on trial runs and on the preceding factors such as GC overhead.

When running concurrent queries, consider as follows:

- Start with 30 GB per executor and all machine cores.
- Create multiple parallel Spark applications by oversubscribing CPU (around 30% latency improvement).
- Distribute queries across parallel applications.
- Modify size based both on trial runs and on the preceding factors such as GC overhead.

As stated before, it's important to keep monitoring the performance, especially outliers, using the timeline view, SQL graph, job statistics etc. It might be a case where one of the executors is slower than the other, which most frequently happens on large clusters (30+ nodes). What you then might consider is to divide the work into more tasks such that the scheduler can compensate for the slower tasks.

If there is an optimization necessary in relation to the optimization of a job execution, make sure you keep in mind the caching (an example might be using the data twice, but cache it). IF you broadcast variables on all the executors you set up, due to the variables only being serialized once, you'll have faster lookups. In another case you might use the thread pool that runs on the driver, which could result in faster operations for many tasks.

<https://docs.microsoft.com/en-us/azure/synapse-analytics/spark/apache-spark-performance>

Question 77: Skipped

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

A window function enables you to perform a mathematical equation on a set of data that is defined within a window. The mathematical equation is typically an aggregate function; however, instead of applying the aggregate function to all the rows in a table, it is applied to a set of rows that are defined by the window function, and then the aggregate is applied to it.

One of the key components of window functions is the [?] clause. This clause determines the partitioning and ordering of a `rowset` before the associated window function is applied. That is, the [?] clause defines a window or user-specified set of rows within a query result set.

☐ `UNDER`

☒ `OVER`
(Correct)

☐ `HAVING`

☐ `WHERE`

Explanation

A window function enables you to perform a mathematical equation on a set of data that is defined within a window. The mathematical equation is typically an aggregate function; however, instead of applying the aggregate function to all the rows in a table, it is applied to a set of rows that are defined by the window function, and then the aggregate is applied to it.

It is used to either perform calculations against a range of data, but it can also be used to programmatically define a deduplication of data technique, or paginate results.

One of the key components of window functions is the `OVER` clause. This clause determines the partitioning and ordering of a `rowset` before the associated window function is applied. That is, the `OVER` clause defines a window or user-specified set of

rows within a query result set. A window function then computes a value for each row in the window. You can use the `OVER` clause with functions to compute aggregated values such as moving averages, cumulative aggregates, running totals, or a top N per group results.

```
SQL
-- Syntax for SQL Server, Azure SQL Database, and Azure Synapse Analytics

OVER (
[ <PARTITION BY clause> ]
[ <ORDER BY clause> ]
[ <ROW or RANGE clause> ]
)

<PARTITION BY clause> ::=
PARTITION BY value_expression , ... [ n ]

<ORDER BY clause> ::=
ORDER BY order_by_expression
[ COLLATE collation_name ]
[ ASC | DESC ]
[ ,...n ]

<ROW or RANGE clause> ::=
{ ROWS | RANGE } <window frame extent>

<window frame extent> ::=
{ <window frame preceding>
| <window frame between>
}

<window frame between> ::=
BETWEEN <window frame bound> AND <window frame bound>

<window frame bound> ::=
{ <window frame preceding>
| <window frame following>
```

```

}

<window frame preceding> ::=
{
UNBOUNDED PRECEDING
| <unsigned_value_specification> PRECEDING
| CURRENT ROW
}

<window frame following> ::=
{
UNBOUNDED FOLLOWING
| <unsigned_value_specification> FOLLOWING
| CURRENT ROW
}

<unsigned value specification> ::=
{ <unsigned integer literal> }

```

<https://docs.microsoft.com/en-us/sql/t-sql/queries/select-over-clause-transact-sql?view=sql-server-ver15>

You can then use aggregate functions with our window by expanding on our query that uses the OVER clause. The following aggregate functions are supported including COUNT, MAX, AVG, SUM, APPROX_COUNT, DISTINCT, MIN, STDEV, STDEVP, STRING_AGG, VAR, VARP, GROUPING, GROUPING_ID, COUNT_BIG, CHECKSUM_AGG

Alternatively, you can use analytical functions, which calculate an aggregate value based on a group of rows. Unlike aggregate functions, however, analytic functions can return multiple rows for each group. Use analytic functions to compute moving averages, running totals, percentages, or top-N results within a group. Supports LAG, LEAD, FIRST_VALUE, LAST_VALUE, CUME_DIST, PERCENTILE_CONT, PERCENTILE_DISC, PERCENT_RANK

You may want to use the ROWS and RANGE clauses to further limit the rows within the partition by specifying start and end points within the partition. This is done by specifying a range of rows with respect to the current row either by logical association or physical association. Physical association is achieved by using the ROWS clause.

Supports PRECEDING, UNBOUNDING PRECEDING, CURRENT
ROW, BETWEEN, FOLLOWING, UNBOUNDED FOLLOWING

Finally, window functions support Ranking functions like RANK, NTILE, DENSE_RANK, ROW_NUMBER.

<https://docs.microsoft.com/en-us/azure/stream-analytics/stream-analytics-window-functions>

Question 78: Skipped

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

Which of the following are valid options for transforming data within Azure Data Factory? (Select three)

- ☐

Data Storage Activities

- ☐ Analytic Flows
- ☐ Compute Resources
(Correct)
- ☐ Test Lab Packages
- ☐ Control Resources
- ☐ Mapping Data Flows
(Correct)
- ☐ SSIS Packages
(Correct)
- ☐ Data Movement Flows

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

Transforming data using compute resources

Azure Data Factory can also call on compute resources to transform data by a data platform service that may be better suited to the job. A great example of this is that Azure Data Factory can create a pipeline to an analytical data platform such as Spark pools in an Azure Synapse Analytics instance to perform a complex calculation using python. Another example could be to send data to an Azure SQL Database instance to execute a stored procedure using Transact-SQL. There is a wide range of compute

resource, and the associated activities that they can perform as shown in the following table:

Compute environment: On-demand HDInsight cluster or your own HDInsight cluster

Activities: Hive, Pig, Spark, MapReduce, Hadoop Streaming

Compute environment: Azure Batch

Activities: Custom activities

Compute environment: Azure Machine Learning Studio Machine

Activities: Learning activities: Batch Execution and Update Resource

Compute environment: Azure Machine Learning

Activities: Azure Machine Learning Execute Pipeline

Compute environment: Azure Data Lake Analytics

Activities: Data Lake Analytics U-SQL

Compute environment: Azure SQL, Azure SQL Data Warehouse, SQL Server

Activities: Stored Procedure

Compute environment: Azure Databricks

Activities: Notebook, Jar, Python

Compute environment: Azure Function

Activities: Azure Function activity

Transforming data using SQL Server Integration Services (SSIS) packages

Many organizations have decades of development investment in SSIS packages that contain both ingestion and transformation logic from on-premises and cloud data stores. Azure Data Factory provides the ability to lift and shift existing SSIS workload, by creating an Azure-SSIS Integration Runtime to natively execute SSIS packages. Using Azure-SSIS Integration Runtime will enable you to deploy and manage your existing SSIS packages with little to no change using familiar tools such as SQL Server Data Tools (SSDT) and SQL Server Management Studio (SSMS), just like using SSIS on premises.

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

Question 79: Skipped

What is Apache Spark notebook?

- ☐ A cloud-based Big Data and Machine Learning platform, which empowers developers to accelerate AI and innovation by simplifying the process of building enterprise-grade production data applications.
- ☐ 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.
- ☐ The logical Azure Databricks environment in which clusters are created, data is stored (via DBFS), and in which the server resources are housed.
- ☐

A notebook is a collection of cells. These cells are run to execute code, to render formatted text, or to display graphical visualizations.

(Correct)

Explanation

What is Apache Spark notebook?

A notebook is a collection of cells. These cells are run to execute code, to render formatted text, or to display graphical visualizations.

What is a cluster?

The notebooks are backed by clusters, or networked computers, that work together to process your data. The first step is to create a cluster.

<https://azure-ramitgridhar.blogspot.com/2019/07/azure-databricks-create-cluster-and.html>

Question 80: Skipped

When loading data into Azure Synapse Analytics on a scheduled basis, it's important to try to reduce the time taken to not perform the data load, and minimize the resources needed as much as possible to maintain good performance cost-effectively.

Which of the following are valid Strategies for managing source data files? (Select all that apply)

- ☒ When loading large datasets, it's best to use the compression capabilities of the file format.
(Correct)
- ☒ Maintaining a well-engineered Data Lake structure
(Correct)
- ☐ Consolidate source files



Having well defined “zones” established for the data coming into the Data Lake and cleansing and transformation tasks that land the data you need in a curated and optimized state.

(Correct)

Explanation

When loading data into Azure Synapse Analytics on a scheduled basis, it's important to try to reduce the time taken to not perform the data load, and minimize the resources needed as much as possible to maintain good performance cost-effectively.

Strategies for managing source data files include:

Maintain a well-engineered data lake structure

Maintaining a well-engineered Data Lake structure allows you to know that the data your loading regularly is consistent with the data requirements for your system. It is less important if your load is a once-off or exploratory rather than analytical. Some strategies include folder hierarchies based on the source system, and date/time or file format and focus.

In general, having well defined “zones” established for the data coming into the Data Lake and cleansing and transformation tasks that land the data you need in a curated and optimized state.

Compress and optimize files

When loading large datasets, it's best to use the compression capabilities of the file format. It ensures that less time is spent on the process of data transfers, using instead the power of Azure Synapse' Massively Parallel Processing (MPP) compute capabilities for decompression.

It is fairly standard to maintain curated source files in columnar compressed file formats such as RC, Gzip, Parquet, and ORC, which are all supported import formats.

Split source files

One of the key architectural components within Azure Synapse Analytics dedicated SQL pools is the decoupled storage that is segmented into 60 parts. You should maintain alignment to multiples of this number as much as possible depending on the file sizes that you are loading, and the number of compute nodes you have provisioned. Since there are 60 storage segments and a maximum of 60 MPP compute nodes within the highest performance configuration of SQL Pools, a 1:1 file to compute node to storage segment may be viable for ultra-high workloads, reducing the load times to the minimum possible.

<https://docs.microsoft.com/en-us/azure/synapse-analytics/sql/data-processed>

Question 81: Skipped

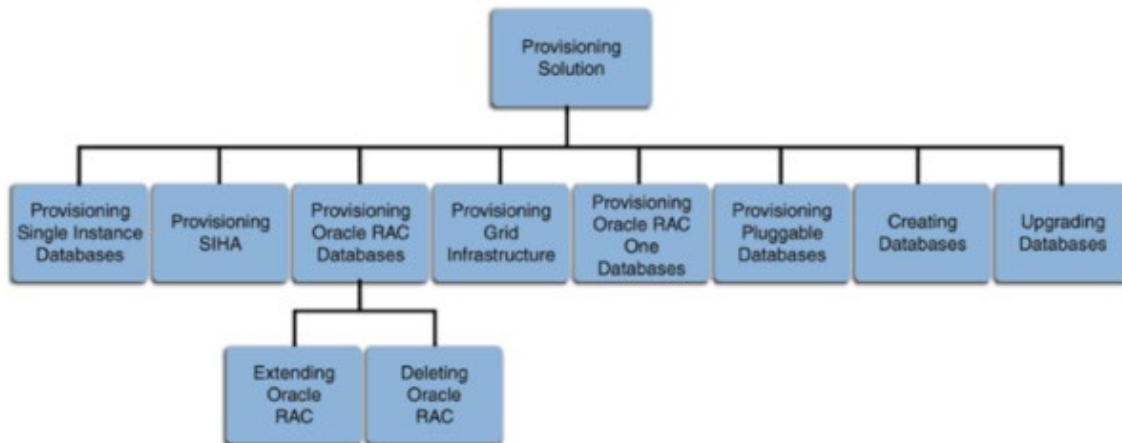
Identify the missing word(s) in the following sentence within the context of Microsoft Azure.

The act of setting up the database server is called [?].

- ☒ Provisioning
(Correct)
- ☐ Distribution
- ☐ Running up
- ☐ Population

Explanation

The act of setting up the database server is called *provisioning*.



https://docs.oracle.com/cd/E24628_01/em.121/e27046/prov_db_overview.htm#EMLCM11094

Question 82: Skipped

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

Azure Storage provides a REST API to work with the containers and data stored in each account. The simplest way to handle access keys and endpoint URLs within applications is to use [?].

- ☐ The account subscription key
- ☐ The private access key
- ☐ The instance key
- ☐ The REST API endpoint
- ☐ A public access key
- ☒ Storage account connection strings
(Correct)

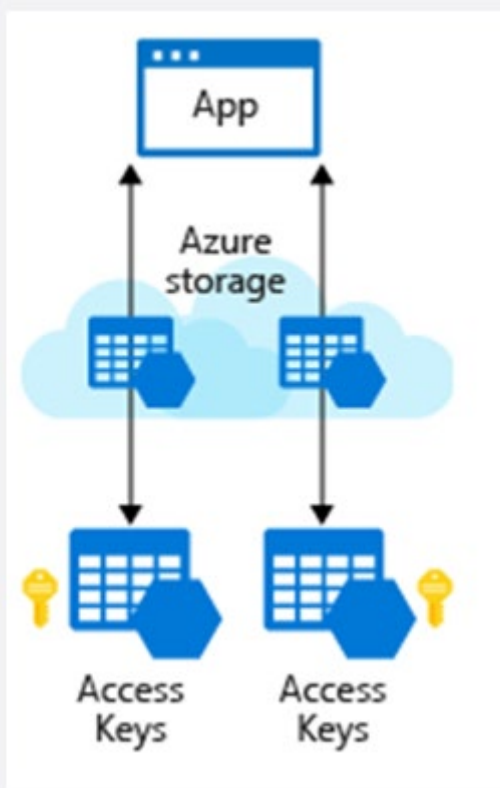
Explanation

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

- Access key
- REST API endpoint

Security access keys

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



Connection strings

The simplest way to handle access keys and endpoint URLs within applications is to use **storage account connection strings**. A connection string provides all needed connectivity information in a single text string.

Azure Storage connection strings look similar to the following example, but with the access key and account name of your specific storage account:

```
DefaultEndpointsProtocol=https;AccountName={your-storage};  
AccountKey={your-access-key};  
EndpointSuffix=core.windows.net
```

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

Question 83: Skipped

Activities within Azure Data Factory define the actions that will be performed on the data. Which are valid activity categories? (Select three)

- ☒ Control activities
(Correct)
- ☒ Data movement activities
(Correct)
- ☐ Analytic activities
- ☒ Data transformation activities
(Correct)
- ☐ Test Lab activities
- ☐ Data storage activities

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

Data movement activities

Data movement activities simply move data from one data store to another. You can use the Copy Activity to perform data movement activities, or by using JSON. There are a wide range of data stores that are supported as a source and as a sink. This list is ever increasing, and you can find the latest information

here: <https://docs.microsoft.com/en-us/azure/data-factory/concepts-pipelines-activities#data-movement-activities>

Data transformation activities

Data transformation activities can be performed natively within the authoring tool of Azure Data Factory using the Mapping Data Flow. Alternatively, you can call a compute resource to change or enhance data through transformation, or perform analysis of the data. These include compute technologies such as Azure Databricks, Azure Batch, SQL Database and Azure Synapse Analytics, Machine Learning Services, Azure Virtual machines and HDInsight. You can make use of any existing SQL Server Integration Services (SSIS) Packages stored in a catalogue to execute in Azure

As this list is always evolving, you can get the latest information here: <https://docs.microsoft.com/en-us/azure/data-factory/concepts-pipelines-activities#data-transformation-activities>

Control activities

When graphically authoring ADF solutions, you can use the control flow within the designed to orchestrate pipeline activities that include chaining activities in a sequence, branching, defining parameters at the pipeline level, and passing arguments while invoking the pipeline on-demand or from a trigger. The current capabilities include:

- **Execute Pipeline Activity**

Execute Pipeline activity allows a Data Factory pipeline to invoke another pipeline.

- **ForEachActivity**

ForEach Activity defines a repeating control flow in your pipeline. This activity is used to iterate over a collection and executes specified activities in a loop. The loop implementation of this activity is similar to Foreach looping structure in programming languages.

- **WebActivity**

Web Activity can be used to call a custom REST endpoint from a Data Factory pipeline. You can pass datasets and linked services to be consumed and accessed by the activity.

- **Lookup Activity**

Lookup Activity can be used to read or look up a record/ table name/ value from any external source. This output can further be referenced by succeeding activities.

- Get Metadata Activity

GetMetadata activity can be used to retrieve metadata of any data in Azure Data Factory.

- Until Activity

Implements Do-Until loop that is similar to Do-Until looping structure in programming languages. It executes a set of activities in a loop until the condition associated with the activity evaluates to true. You can specify a timeout value for the until activity in Data Factory.

- If Condition Activity

The If Condition can be used to branch based on condition that evaluates to true or false. The If Condition activity provides the same functionality that an if statement provides in programming languages. It evaluates a set of activities when the condition evaluates to true and another set of activities when the condition evaluates to false.

- Wait Activity

When you use a Wait activity in a pipeline, the pipeline waits for the specified period of time before continuing with execution of subsequent activities.

You can get the latest information here: <https://docs.microsoft.com/en-us/azure/data-factory/concepts-pipelines-activities#control-activities>

Question 84: Skipped

In Azure Synapse Studio, manage integration pipelines within the Integrate hub.

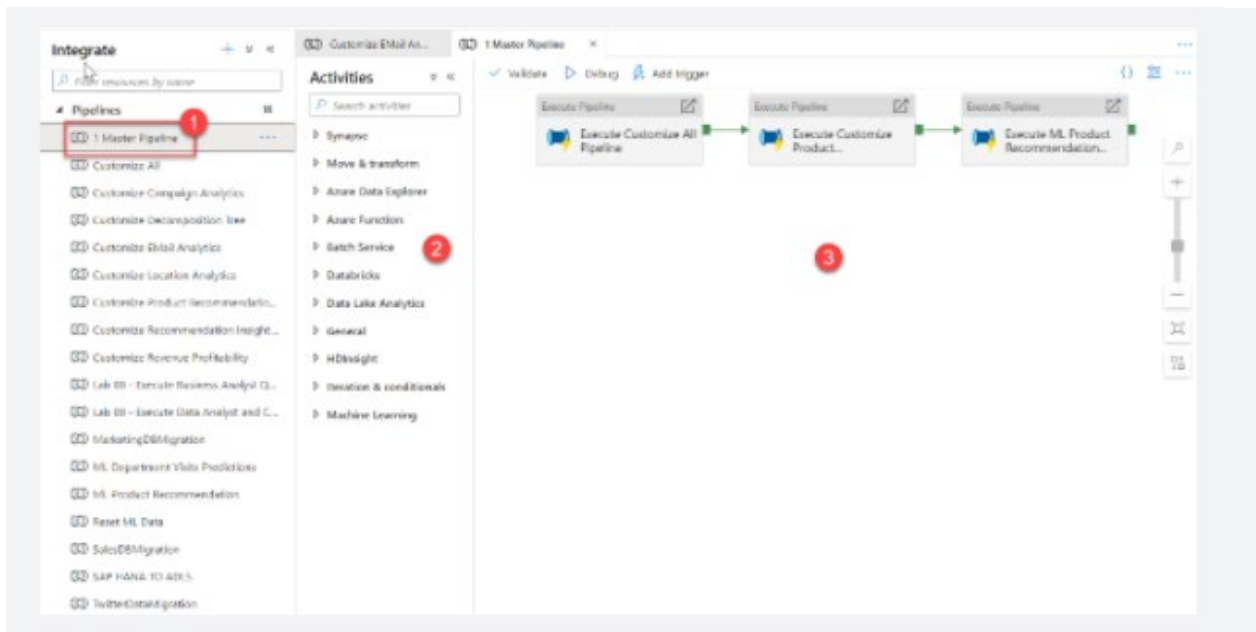
When you expand Pipelines you will see which of the following? (Select three)

- ☐ Data flows
- ☐ Notebooks
- ☒ Pipeline canvas
(Correct)
- ☒ Master Pipeline
(Correct)
- ☐ SQL serverless databases
- ☒ Activities
(Correct)
- ☐ SQL scripts
- ☐ Power BI
- ☐ Provisioned SQL pool databases
- ☐ External data sources

Explanation

In Azure Synapse Studio, manage integration pipelines within the Integrate hub. If you are familiar with Azure Data Factory, then you will feel at home in this hub. The pipeline creation experience is the same as in ADF, which gives you another powerful integration built into Synapse Analytics, removing the need to use Azure Data Factory for data movement and transformation pipelines.

When you expand Pipelines you will see **Master Pipeline (1)**. Point out the **Activities (2)** that can be added to the pipeline, and show the **pipeline canvas (3)** on the right.



This Synapse workspace contains 16 pipelines that enable us to orchestrate data movement and transformation steps over data from several sources.

The **Activities** list contains many activities that you can drag and drop onto the pipeline canvas on the right.

Expand a few activity categories to show what's available, such as Notebook, Spark, and SQL pool stored procedure activities under Synapse.

<https://techcommunity.microsoft.com/t5/azure-synapse-analytics/quickly-get-started-with-azure-synapse-studio/ba-p/1961116>

Question 85: Skipped

Azure Synapse Pipelines is the cloud-based ETL and data integration service that allows you to create data-driven workflows for orchestrating data movement and transforming data at scale.

Azure Synapse Pipelines enables you to integrate data pipelines between which of the following? (Select all that apply)

- ☒ SQL Serverless
(Correct)
- ☒ SQL Pools
(Correct)
- ☐ Hadoop Pools
- ☒ Spark Pools
(Correct)
- ☐ Cosmos Pools
- ☐ Cosmos Serverless

Explanation

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

Much of the functionality of Azure Synapse Pipelines come from the Azure Data Factory features and are commonly referred to as Pipelines. Azure Synapse Pipelines enables you to integrate data pipelines between SQL Pools, Spark Pools and SQL Serverless, providing a one stop shop for all your analytical needs.

Like Azure Data Factory, Azure Synapse Pipelines is composed of four core components. These components work together to provide the platform on which you can compose data-driven workflows with steps to move and transform data.

Azure Data Factory Components



Data Factory supports a wide variety of data sources that you can connect to through the creation of an object known as a **Linked Service**, which enables you to ingest the data from a data source in readiness to prepare the data for transformation and/or analysis. In addition, Linked Services can fire up compute services on demand. For example, you may have a requirement to start an on-demand HDInsight cluster for the purpose of just processing data through a Hive query. So Linked Services enables you to define data sources, or compute resource that is required to ingest and prepare data.

With the linked service defined, Azure Data Factory is made aware of the datasets that it should use through the creation of a **Datasets** object. Datasets represent data structures within the data store that is being referenced by the Linked Service object. Datasets can also be used by an ADF object known as an Activity.

Activities typically contain the transformation logic or the analysis commands of the Azure Data Factory's work. Activities includes the Copy Activity that can be used to ingest data from a variety of data sources. It can also include the Mapping Data Flow to perform code-free data transformations. It can also include the execution of a stored procedure, Hive Query, or Pig script to transform the data. You can push data into a Machine Learning model to perform analysis. It is not uncommon for multiple activities to take place that may include transforming data using a SQL stored procedure and then perform analytics with Databricks. In this case, multiple activities can be logically grouped together with an object referred to as a **Pipeline**, and these can be *scheduled* to execute, or a *trigger* can be defined that determines when a pipeline execution needs to be kicked off. There are different types of triggers for different types of events.



Control flow is an orchestration of pipeline activities that includes chaining activities in a sequence, branching, defining parameters at the pipeline level, and passing arguments while invoking the pipeline on-demand or from a trigger. It also includes custom-state passing and looping containers, and For-each iterators.

Parameters are key-value pairs of read-only configuration. Parameters are defined in the pipeline. The arguments for the defined parameters are passed during execution from the run context that was created by a trigger or a pipeline that was executed manually. Activities within the pipeline consume the parameter values.

Azure Synapse pipelines has an *integration runtime* that enables it to bridge between the activity and linked Services objects. It is referenced by the linked service, and provides the compute environment where the activity either runs on or gets dispatched from. This way, the activity can be performed in the region closest possible. There are three types of Integration Runtime, including Azure, Self-hosted, and Azure-SSIS.

Once all the work is complete, you can then use Data Factory to publish the final dataset to another linked service that can then be consumed by technologies such as Power BI or Machine Learning.

<https://docs.microsoft.com/en-us/azure/synapse-analytics/sql-data-warehouse/sql-data-warehouse-partner-data-integration>

Question 86: Skipped

Setting Global parameters in an Azure Data Factory pipeline, allows you to use constants for consumption in pipeline expressions.

If you have created a data flow in which you have set parameters, it is possible to execute it from a pipeline using the Execute Data Flow Activity. Once you have added

the activity to the pipeline canvas, you'll find the data flow parameters in the activity's Parameters tab.

Is it possible to combine the pipeline and data flow expression parameters while mapping dataflow?

☐ No

☒ Yes

(Correct)

Explanation

Global parameters in Azure Data Factory

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

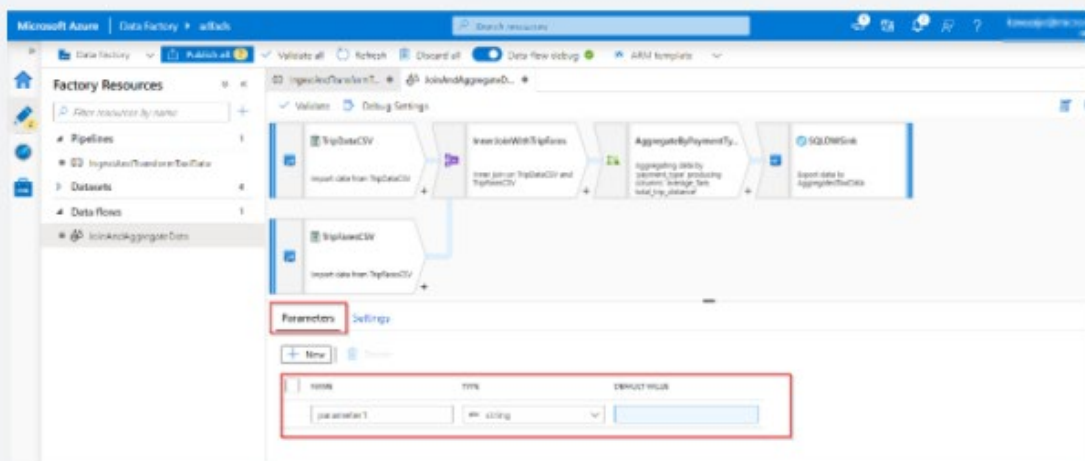
Using global parameters in a pipeline

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

Create parameters in dataflow

To add parameters to your data flow, click on the blank portion of the data flow canvas to see the general properties. In the settings pane, you will see a tab called Parameter.

Select New to generate a new parameter. For each parameter, you must assign a name, select a type, and optionally set a default value.



Assign parameters from a pipeline in mapping dataflow

If you have created a data flow in which you have set parameters, it is possible to execute it from a pipeline using the Execute Data Flow Activity. Once you have added the activity to the pipeline canvas, you'll find the data flow parameters in the activity's Parameters tab. Assigning parameter values, ensures that you are able to use the parameters in a pipeline expression language or data flow expression language based on spark types. **You can also combine the two, that is, pipeline and data flow expression parameters.**

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

Question 87: Skipped

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

All data within an Azure Cosmos DB container is partitioned based on the [?], and applies to both the transactional store and the analytical store. Boundaries for parallelizing workloads are based on this [?].

- ☐ Index key
- ☒ Partition key
(Correct)
- ☐ Primary key
- ☐ Foreign key

Explanation

Mixed entity types per container

You may want to mix different document entity types (entities) in the same container, which is useful to efficiently retrieve data for both entities using a single query. For example, you could put both customer profile and sales order data in the same container and partition it by customerId. In such a situation, you would usually add a field to your documents that identifies the entity type of each document to differentiate between them at query time. In the following sample documents, you will see that the type is added for this purpose in the following example documents:

```
JSON
{
  "id": "54AB87A7-BDB9-4FAE-A668-AA9F43E26628",
  "type": "customer",
  "name": "Franklin Ye",
  "customerId": "54AB87A7-BDB9-4FAE-A668-AA9F43E26628",
  "address": {
    "streetNo": 15850,
    "streetName": "NE 40th St.",
    "postcode": 98052
  }
}

{
```

```
{
  "_id": "000C23D8-B8BC-432E-9213-6473DFDA2BC5",
  "type": "salesOrder",
  "customerId": "54AB87A7-BDB9-4FAE-A668-AA9F43E26628",
  "orderDate": "2014-02-16T00:00:00",
  "shipDate": "2014-02-23T00:00:00",
  "details": [
    {
      "sku": "BK-R64Y-42",
      "name": "Road-550-W Yellow, 42",
      "price": 1120.49,
      "quantity": 1
    }
  ]
}
```

The following query on against the transactional store would return the customer details and all orders associated with this one customer.

SQL

```
SELECT * FROM c WHERE c.customerID = "54AB87A7-BDB9-4FAE-A668-AA9F43E26628"
```

Whilst this approach to modelling is potentially useful for your Cosmos DB transactional store queries. All documents within a single container are mapped to a single analytical store, leading to sparsely populated column stores with the different data types needing to be further separated at the time of running an analytical query.

Recommendation: As with many design decisions, there is a trade-off between the efficiency of querying the transactional store and the ease of querying the analytical store. Carefully evaluate the usefulness of storing a mix of different document entity types in the same container to your transactional workloads. If you choose to do so, you will be required to filter by the property entity type property you selected.

Embedding entity arrays

When optimizing transactional data models, we choose to embed entities within an array in a document, especially for read heavy workloads where:

- There are contained relationships between entities.
- There are one-to-few relationships between entities.
- There is embedded data that changes infrequently.
- There is embedded data that will not grow without bound.

- There is embedded data that is queried frequently together.

Due to the fact that there are one to few relationships between the embedded entities that are represented within a single document, and that these are mapped to a single column within a single row within the analytical store. The entire embedded entity array will reside within a single column value, and need to be translated from its JSON representation at the time of querying in order to retrieve embedded entity values, irrespective of which of the two modes of schema representation being used.

Recommendation: Again, a balance needs to be struck between the usefulness of the entity embedding within the transactional application and the added complexity of writing queries against embedded JSON documents for your application.

Partitioning of data

All data within an Azure Cosmos DB container is partitioned based on the partition key, and applies to both the transactional store and the analytical store. Boundaries for parallelizing workloads are based on this partition key.

The orderliness associated when data appears in the analytical store for a query is only guaranteed within a partition. As an example, when documents (1) (2) (3) are inserted in the transactional store into a single partition, they are guaranteed to be present in the analytical store in the order in which they were inserted.

<https://docs.microsoft.com/en-us/azure/cosmos-db/modeling-data>

Question 88: Skipped

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

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

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

ADLS Passthrough is configured when you create a cluster in the Azure Databricks workspace. On a *standard cluster*, when you enable this setting ... [?]

- ☐ you will inherit user access from the Azure Active Directory (AAD) users to the Azure Databricks workspace.
- ☐ you must set two user accesses to one of the Azure Active Directory (AAD) users in the Azure Databricks workspace. The second is required as a backup or secondary user.
- ☐ you may set multiple user accesses to one of the Azure Active Directory (AAD) users in the Azure Databricks workspace. The additional access are required as a backup or auxiliary users.
- ☒ you must set single user access to one of the Azure Active Directory (AAD) users in the Azure Databricks workspace.
(Correct)

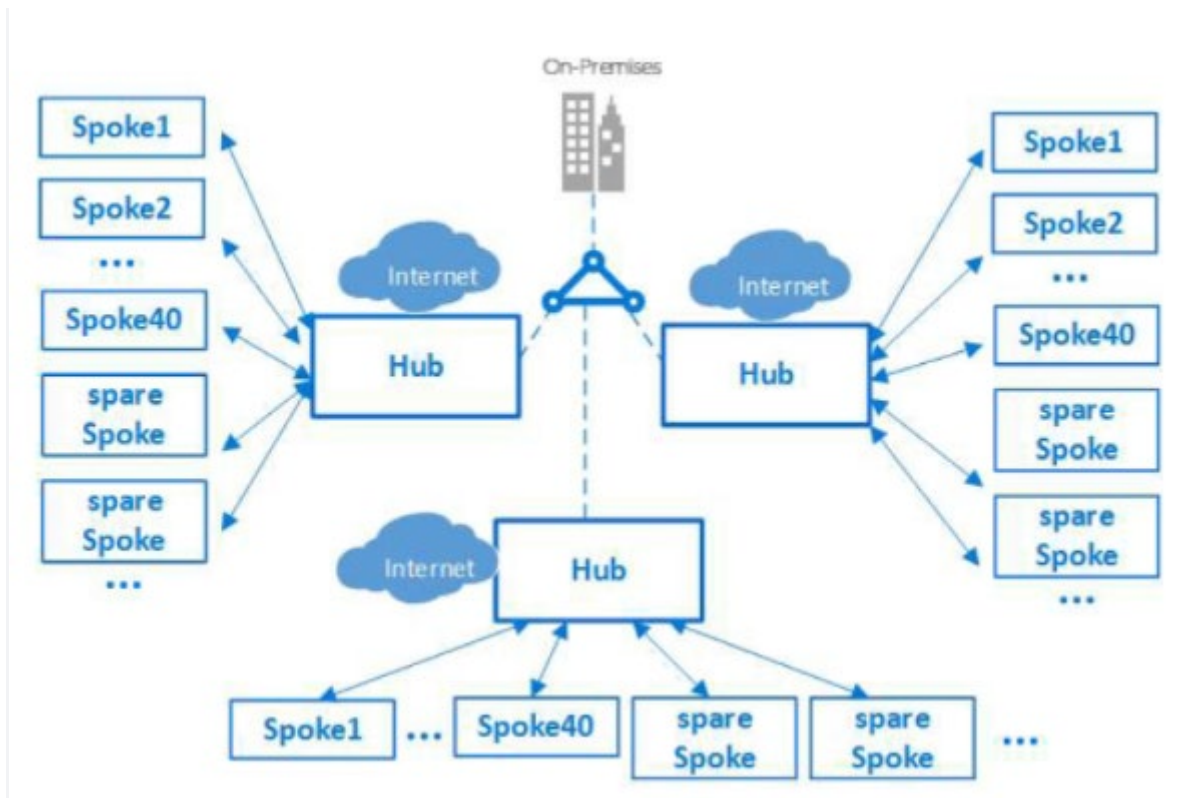
Explanation

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

Consider isolating each workspace in its own VNet

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

1. Deploy each Workspace in its own spoke VNet.
2. Put all the common networking resources in a central hub VNet, such as your custom DNS server.
3. Join the Workspace spokes with the central networking hub using [VNet Peering](#)



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

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

1. The lifecycle of default DBFS is tied to the Workspace. Deleting the workspace will also delete the default DBFS and permanently remove its contents.
2. One can't restrict access to this default folder and its contents.

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

Always hide secrets in a key vault

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

If using Azure Key Vault, create separate AKV-backed secret scopes and corresponding AKVs to store credentials pertaining to different data stores. This will

help prevent users from accessing credentials that they might not have access to. Since access controls are applicable to the entire secret scope, users with access to the scope will see all secrets for the AKV associated with that scope.

Access control - Azure Data Lake Storage (ADLS) passthrough

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

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

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

Azure Data Lake Storage Credential Passthrough ?

☒ Enable credential passthrough for user-level data access

Single User Access ?

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

▼ Advanced Options

Azure Data Lake Storage Credential Passthrough ?

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

Configure audit logs and resource utilization metrics to monitor activity

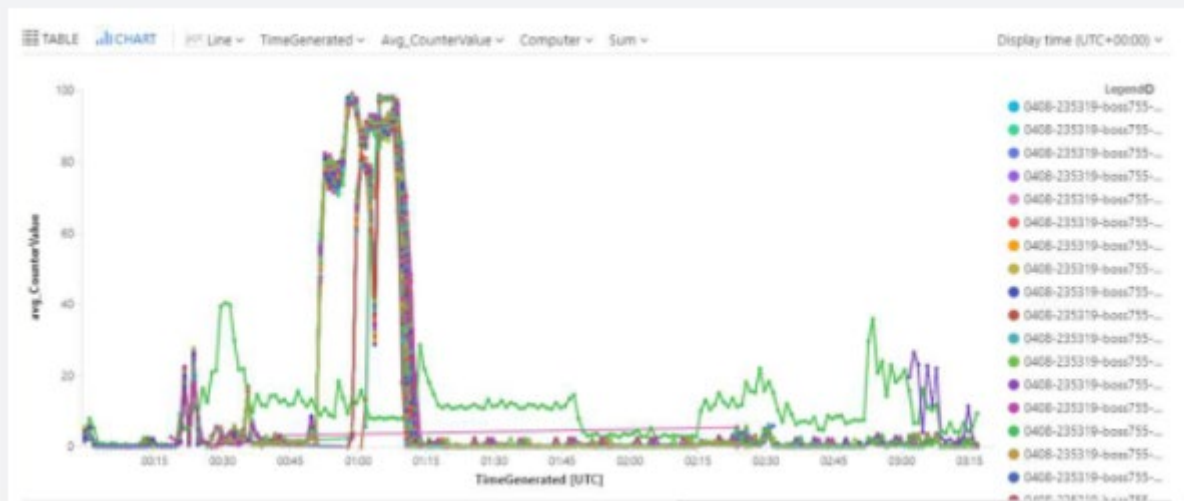
An important facet of monitoring is understanding the resource utilization in Azure Databricks clusters. You can also extend this to understanding utilization across all clusters in a workspace. This information is useful in arriving at the correct cluster and VM sizes. Each VM does have a set of limits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

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

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

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

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



1. <https://docs.microsoft.com/azure/azure-monitor/learn/quick-collect-linux-computer>
 2. <https://github.com/Microsoft/OMS-Agent-for-Linux/blob/master/docs/OMS-Agent-for-Linux.md>
 3. <https://github.com/Microsoft/OMS-Agent-for-Linux/blob/master/docs/Troubleshooting.md>
-

Question 89: 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 Tungsten which 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.

When we shuffle data, it creates what is known as [?].

- ☐ A Stage
- ☐ A Lineage
- ☒ A Stage boundary
(Correct)
- ☐ A Pipeline

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.

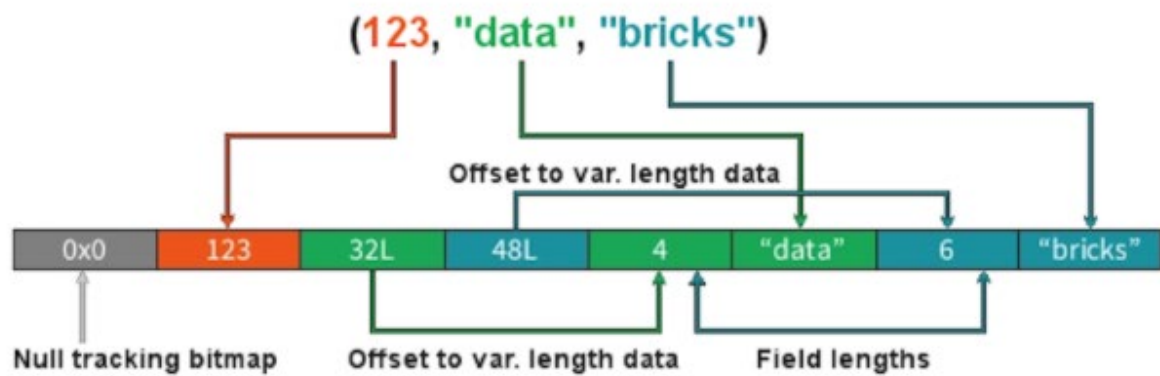
`UnsafeRow` is the in-memory storage format for Spark SQL, DataFrames & Datasets.

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.

Why Work Backwards?

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>

What happens to Databricks activities (notebook, JAR, Python) in Azure Data Factory if the target cluster in Azure Databricks isn't running when the cluster is called by Data Factory?

- ☒ If the target cluster is stopped, Databricks will start the cluster before attempting to execute.
(Correct)
- ☐ The Databricks activity will fail in Azure Data Factory – you must always have the cluster running.
- ☐ Whenever a cluster is paused or shut down, ADF will recover from the last operational PiT.
- ☐ Simply add a Databricks cluster start activity before the notebook, JAR, or Python Databricks activity.

Explanation

This situation will result in a longer execution time because the cluster must start, but the activity will still execute as expected.

<https://docs.microsoft.com/en-us/azure/data-factory/transform-data-databricks-python>