Question 51: Skipped

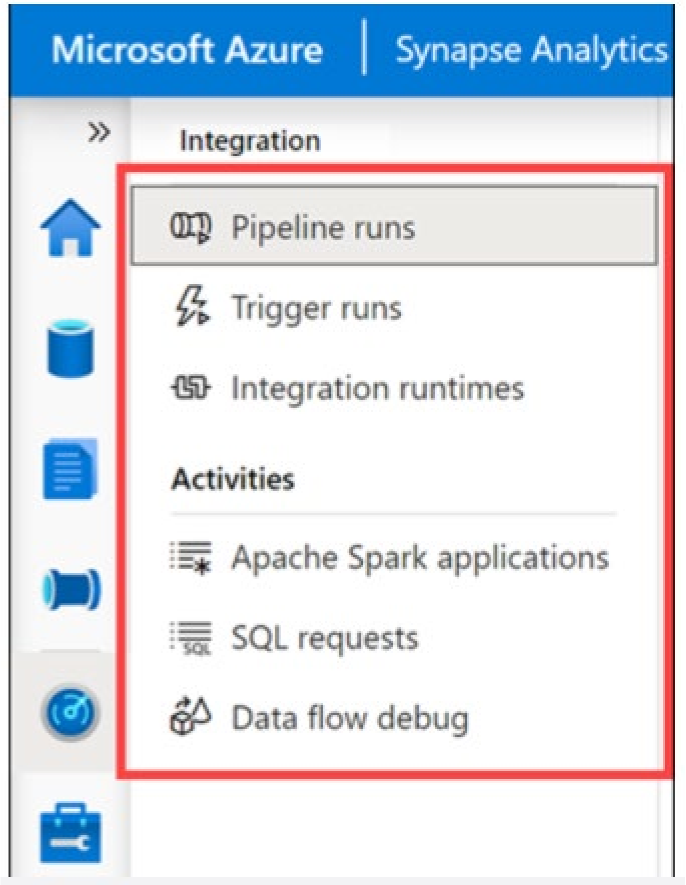
In Azure Synapse Studio, use the Monitor hub is where you access which of the following? (Select six)

- □
  SQL requests
  (Correct)
- □
  SQL serverless databases
- □
  Integration runtimes
  (Correct)
- □
  External data sources
- □
  Data flows
- □
  Pipeline runs
  (Correct)
- □
  Provisioned SQL pool databases
- □
  Notebooks
- □
  Trigger runs
  (Correct)
- □
  Power BI
- □
  Data flow debug
  (Correct)
- □
  Apache Spark jobs
  (Correct)

Explanation

In Azure Synapse Studio, use the Monitor hub to view pipeline and trigger runs, view the status of the various integration runtimes that are running, view Apache Spark jobs, SQL requests, and data flow debug activities.

The Monitor hub is your first stop for debugging issues and gaining insight on resource usage. You can see a history of all the activities taking place in the workspace and which ones are active now.



https://techcommunity.microsoft.com/t5/azure-synapse-analytics/explore-the-monitor-hub-in-synapse-studio-to-keep-track-of-all/ba-p/1987405

Question 52: **Skipped**

Which of the below have the following characteristics?

• Provide undoubtedly the most well-understood model for holding data.

• The simplest structure of columns and tables makes them very easy to use initially, but the inflexible structure can cause some problems.

• We can communicate with relational databases using SQL.

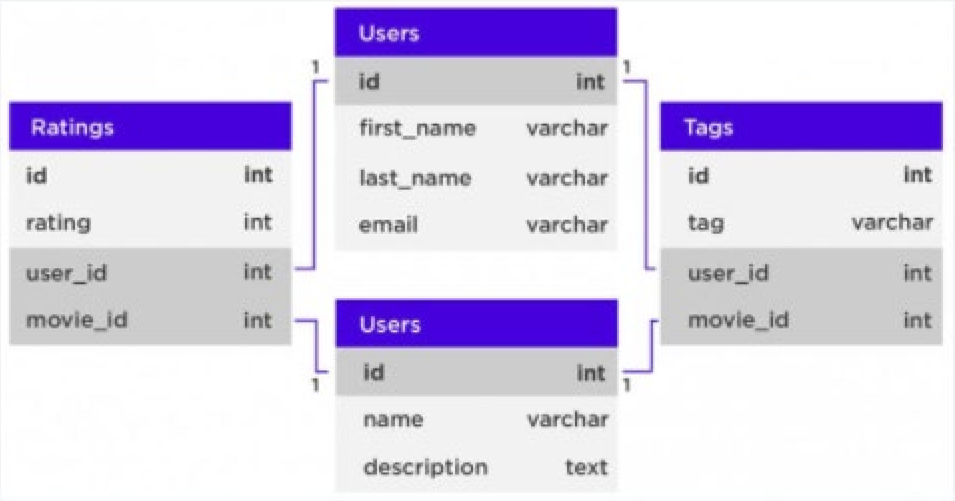
- ○ JSON
- ○ Key-Value
- ○ Non-Relational
- ○ Relational **(Correct)**

Explanation

Relational Data

• Relational databases provide undoubtedly the most well-understood model for holding data.

• The simplest structure of columns and tables makes them very easy to use initially, but the inflexible structure can cause some problems.



Ratings	
id	int
rating	int
user_id	int
movie_id	int

Users	
id	int
first_name	varchar
last_name	varchar
email	varchar

Tags	
id	int
tag	varchar
user_id	int
movie_id	int

Users	
id	int
name	varchar
description	text

- We can communicate with relational databases using **Structured Query Language (SQL).**

- SQL allows the joining of tables using a few lines of code, with a structure most beginner employees can learn very fast.

- Examples of relational databases:
  - MySQL
  - PostgreSQL
  - Db2
  - SQL Server

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

Question 53: Skipped

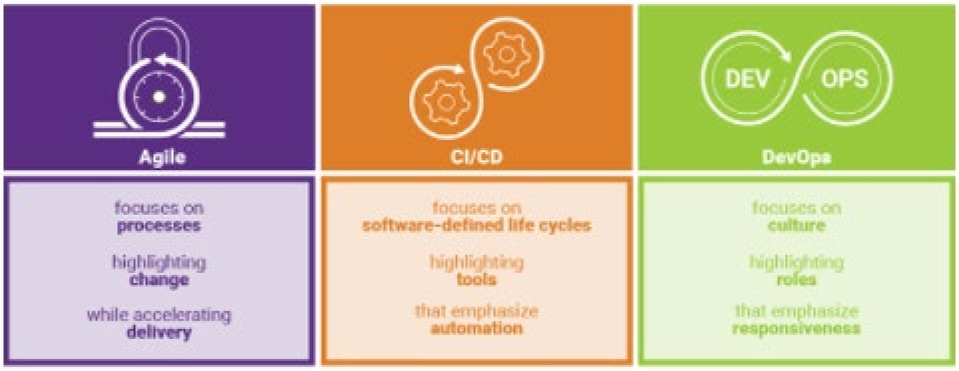
While Agile, CI/CD, and DevOps are different, they support one another

What does CI/CD focus on?

- ○ Culture
- ○ Practices (Correct)
- ○ Strategy
- ○ Development process

Explanation

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


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

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

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

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

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

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

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

CI/CD with Azure DevOps

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

- Integrated Git repositories

- Integration with other Azure services

- Automatic virtual machine management for testing builds

- Secure deployment

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

But what is CI/CD?

Continuous Integration

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

Continuous Delivery

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

Continuous Deployment

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

Who benefits?

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

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

• Data scientists can update models being used in production.

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

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

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

Question 54: Skipped

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

Azure Databricks is a fully-managed, 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. Built as a joint effort by Databricks and Microsoft, Azure Databricks provides data science and engineering teams with a single platform for Big Data processing and Machine Learning.

A Microsoft-managed Azure Databricks workspace virtual network (VNet) exists within the customer subscription. Information exchanged between this VNet and the Microsoft-managed Azure Databricks Control Plane VNet is sent over a secure TLS connection through ports (22 and 5557) that are enabled by Network Security Groups (NSGs) and protected with port IP filtering.

The Blob Storage account provides default file storage within the workspace (databricks file system (DBFS)). This resource and all other Microsoft-managed resources are completely locked from changes made by the customer.

True or False: You can write to the default DBFS file storage as needed, but you cannot change the Blob Storage account settings.

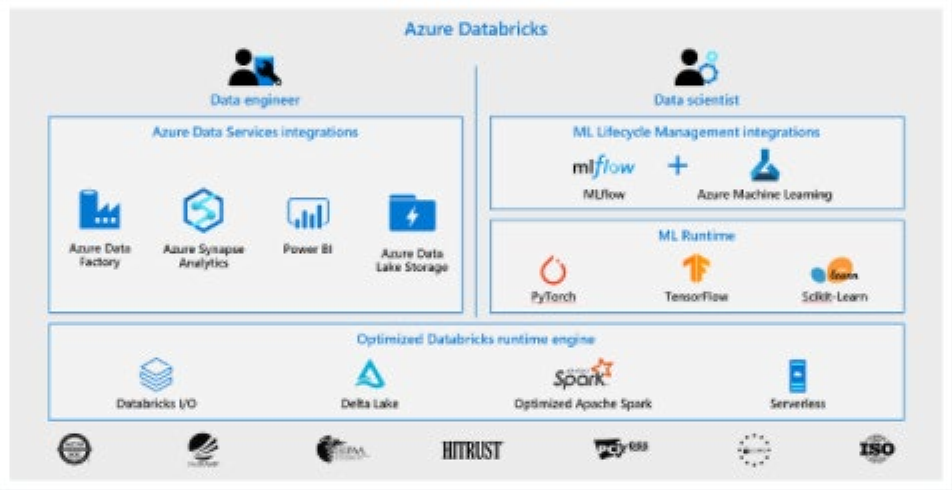
- ○ True (Correct)
- ○ False

Explanation

Azure Databricks is a fully-managed, 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. Built as a joint effort by Databricks and Microsoft, Azure Databricks provides data science and engineering teams with a single platform for Big Data processing and Machine Learning.

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

Conceptual view of Azure Databricks



To provide the best platform for data engineers, data scientists, and business users, Azure Databricks is natively integrated with Microsoft Azure, providing a "first party" Microsoft service. The Azure Databricks collaborative workspace enables these teams to work together through features such as user management, git source code repository integration, and user workspace folders.

Microsoft is working to integrate Azure Databricks closely with all features of the Azure platform. Below is a list of some of the integrations completed so far:

• **VM types**: Many existing VMs can be used for clusters, including F-series for machine learning scenarios, M-series for massive memory scenarios, and D-series for general purpose.

• **Security and Privacy**: Ownership and control of data is with the customer, and Microsoft aims for Azure Databricks to adhere to all the compliance certifications that the rest of Azure provides.

• **Flexibility in network topology**: Azure Databricks supports deployments into virtual networks (VNETs), which can control which sources and sinks can be accessed and how they are accessed.

• **Orchestration**: ETL/ELT workflows (including analytics workloads in Azure Databricks) can be operationalized using Azure Data Factory pipelines.

• **Power BI**: Power BI can be connected directly to Databricks clusters using JDBC in order to query data interactively at massive scale using familiar tools.

- **Azure Active Directory**: Azure Databricks workspaces deploy into customer subscriptions, so naturally AAD can be used to control access to sources, results, and jobs.

- **Data stores**: Azure Storage and Data Lake Store services are exposed to Databricks users via Databricks File System (DBFS) to provide caching and optimized analysis over existing data. Azure Databricks easily and efficiently uploads results into Azure Synapse Analytics, Azure SQL Database, and Azure Cosmos DB for further analysis and real-time serving, making it simple to build end-to-end data architectures on Azure.

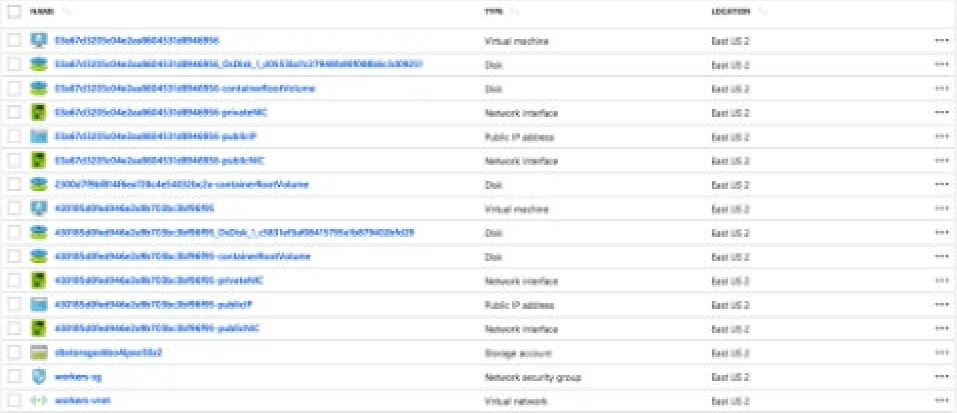
- **Real-time analytics**: Integration with IoT Hub, Azure Event Hubs, and Azure HDInsight Kafka clusters enables developers to build scalable streaming solutions for real-time analytics.

For developers, this design provides three things. First, it enables easy connection to any storage resources in their account, such as an existing Blob storage or Data Lake Store. Second, they are able to take advantage of deep integrations with other Azure services to quickly build data applications. Third, Databricks is managed centrally from the Azure control centre, requiring no additional setup, which allows developers to focus on core business value, not infrastructure management.

Azure Databricks platform architecture

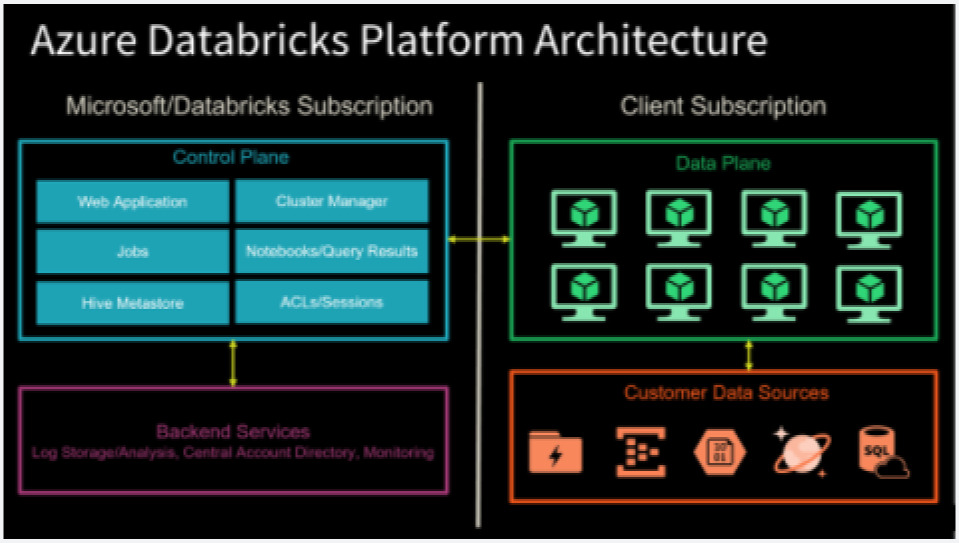
When you create an Azure Databricks service, a "Databricks appliance" is deployed as an Azure resource in your subscription. At the time of cluster creation, you specify the types and sizes of the virtual machines (VMs) to use for both the Driver and Worker nodes, but Azure Databricks manages all other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

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.



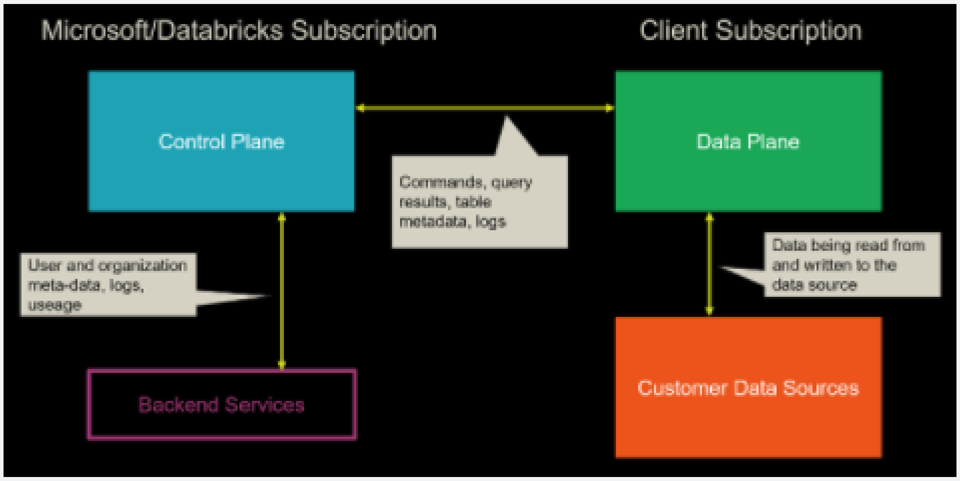
NAME	TYPE	LOCATION	
01a67d3205c04e2aa9604531e8946956	Virtual machine	East US 2	•••
01a67d3205c04e2aa9604531e8946956_OsDisk_1_[illegible]	Disk	East US 2	•••
01a67d3205c04e2aa9604531e8946956-containerRootVolume	Disk	East US 2	•••
01a67d3205c04e2aa9604531e8946956-privateNIC	Network interface	East US 2	•••
01a67d3205c04e2aa9604531e8946956-publicIP	Public IP address	East US 2	•••
01a67d3205c04e2aa9604531e8946956-publicNIC	Network interface	East US 2	•••
2300d7f9b8f14f6ea728c4e54032ba2e-containerRootVolume	Disk	East US 2	•••
430185d0fed946a2a9b703bc8ef96f95	Virtual machine	East US 2	•••
430185d0fed946a2a9b703bc8ef96f95_OsDisk_1_[illegible]	Disk	East US 2	•••
430185d0fed946a2a9b703bc8ef96f95-containerRootVolume	Disk	East US 2	•••
430185d0fed946a2a9b703bc8ef96f95-privateNIC	Network interface	East US 2	•••
430185d0fed946a2a9b703bc8ef96f95-publicIP	Public IP address	East US 2	•••
430185d0fed946a2a9b703bc8ef96f95-publicNIC	Network interface	East US 2	•••
dbstorage[illegible]	Storage account	East US 2	•••
workers-sg	Network security group	East US 2	•••
workers-vnet	Virtual network	East US 2	•••

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 this to further improve Spark performance.

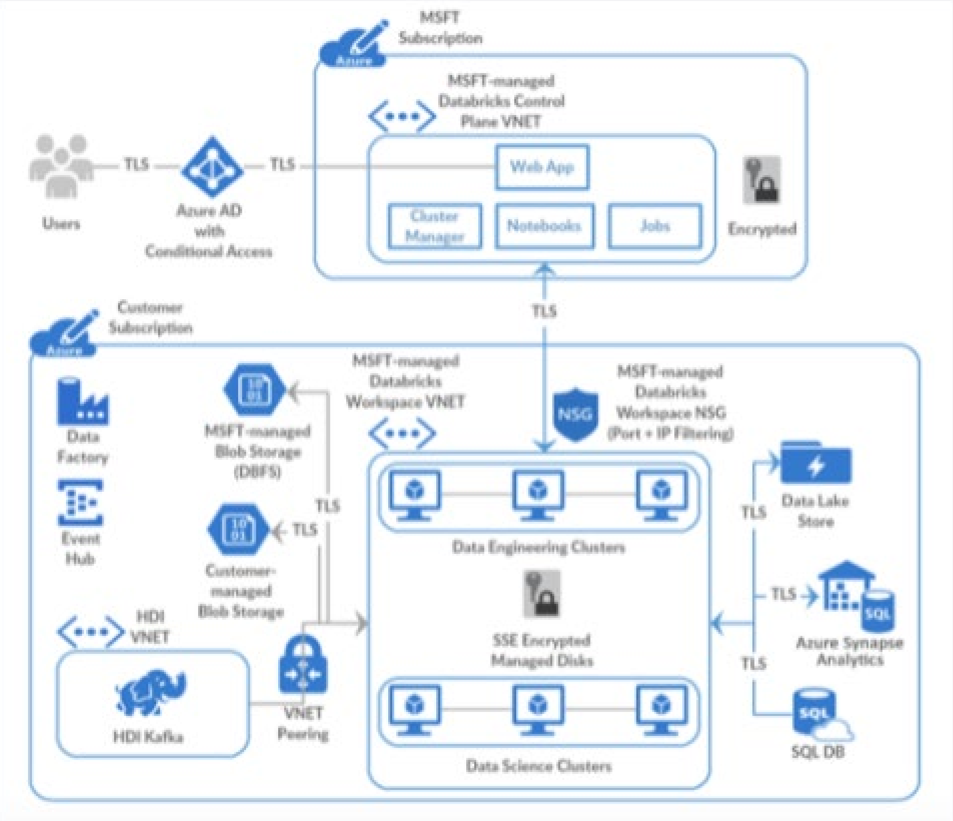


The diagram above shows a Control Plane on the left, which hosts Databricks jobs, notebooks with query results, the cluster manager, web application, Hive metastore, and security access control lists (ACLs) and user sessions. These components are managed by Microsoft in collaboration with Databricks and do not reside within your Azure subscription.

On the right-hand side is the Data Plane, which contains all the Databricks runtime clusters hosted within the workspace. All data processing and storage exists within the client subscription. This means no data processing ever takes place within the Microsoft/Databricks-managed subscription.


Microsoft/Databricks Subscription
Client Subscription
Control Plane
Data Plane
Commands, query results, table metadata, logs
User and organization meta-data, logs, useage
Data being read from and written to the data source
Backend Services
Customer Data Sources

Moving one level deeper, the diagram above shows what is being exchanged between the Azure Databricks platform components. Since the web app and cluster manager is part of the Control Plane, any commands executed in a notebook are sent from the cluster manager to the customer's clusters in the Data Plane. This is because the data processing only occurs within the customer's own subscription, as stated earlier. Any table metadata and logs are exchanged between these two high-level components. Customer data sources within the client subscription exchange data with the Data Plane through read and write activities.


MSFT Subscription
Azure
MSFT-managed Databricks Control Plane VNET
Users
TLS
Azure AD with Conditional Access
TLS
Web App
Cluster Manager
Notebooks
Jobs
Encrypted
TLS
Customer Subscription
Azure
Data Factory
Event Hub
MSFT-managed Blob Storage (DBFS)
MSFT-managed Databricks Workspace VNET
MSFT-managed Databricks Workspace NSG (Port + IP Filtering)
NSG
TLS
TLS
Customer-managed Blob Storage
Data Engineering Clusters
SSE Encrypted Managed Disks
Data Science Clusters
HDI VNET
HDI Kafka
VNET Peering
TLS
Data Lake Store
TLS
Azure Synapse Analytics
SQL
TLS
SQL DB

The diagram above shows a standard deployment that contains the boundaries between the Control Plane and the Data Plane with the Azure components deployed to each. At the top of the diagram is the Control Plane that exists within the Microsoft subscription. The customer subscription is at the bottom of the diagram, which contains the Data Plane and data sources.

A Microsoft-managed Azure Databricks workspace virtual network (VNet) exists within the customer subscription. Information exchanged between this VNet and the Microsoft-managed Azure Databricks Control Plane VNet is sent over a secure TLS connection through ports (22 and 5557) that are enabled by Network Security Groups (NSGs) and protected with port IP filtering.

The Blob Storage account provides default file storage within the workspace (databricks file system (DBFS)). This resource and all other Microsoft-managed resources are completely locked from changes made by the customer. All other resources within the customer subscription are customer-managed and can be added or modified per your Azure subscription permissions. Connectivity between these resources and the Databricks clusters that reside within the Data Plane is secured via TLS.

To clarify, you can write to the default DBFS file storage as needed, but you cannot change the Blob Storage account settings since the account is managed by the Microsoft-managed Control Plane. As a best practice, only use the default storage for temporary files and mount additional storage accounts (Blob Storage or Azure Data Lake Storage Gen2) that you create in your Azure subscription, for long-term file storage. This is because the default file storage is tied to the lifecycle of your Azure Databricks account. If you delete the Azure Databricks account, the default storage gets deleted with it.

If you need advanced network connectivity, such as custom VNet peering and VNet injection, you could deploy Azure Databricks Data Plane resources within your own VNet.

https://docs.databricks.com/getting-started/overview.html

Question 55: Skipped

Before we can query our data using Azure Synapse Analytics using Azure Synapse Link, we must first create the container that is going to hold our data at the same time enabling it to have an analytical store.

True or False: Enabling analytical store is only available at the time of creating a container however it can be deactivated or reactivated at anytime thereafter.

- ○ True
- ○ False (Correct)

Explanation

Before we can query our data using Azure Synapse Analytics using Azure Synapse Link, we must first create the container that is going to hold our data at the same time enabling it to have an analytical store.

Enabling analytical store is only available at the time of creating a container and cannot be completely disabled without deleting the container. Setting the default analytical store TTL value to 0 or null effectively disables the analytical store by no longer synchronize new items to it from the transactional store and deleting items already synchronized from the analytical store.

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

---

Question 56: **Skipped**

With the Azure-SSIS integration runtime installed and SQL Server Data Tools (SSDT) you have the capability to deploy and manage SSIS packages that you create in the cloud. For some packages, you may be able to rebuild them by redeploying them in the Azure-SSIS runtime. However, there may be some SSIS packages that already exist within your environment that may not be compatible.

You can use the [?] to perform an assessment of the SSIS packages that exist and identify any compatibility issues with them.

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

Explanation

With the Azure-SSIS integration runtime installed and SQL Server Data Tools (SSDT) you have the capability to deploy and manage SSIS packages that you create in the cloud. For some packages, you may be able to rebuild them by redeploying them in the Azure-SSIS runtime. However, there may be some SSIS packages that already exist within your environment that may not be compatible? How should you deal with them?

Perform assessments of your SSIS packages.

When you migrate your database workloads from SQL Server on premises to Azure SQL database services, you may have to migrate SSIS packages as well. The first step required is to perform an assessment of you current SSIS packages to make sure that they are compatible in Azure. Fortunately, **you can use the Data Migration Assistant (DMA) to perform an assessment of the SSIS packages that exist and identify any compatibility issues with them.** The Data Migration Assistant has two main categories of information:

- Migration blockers: Issues that prevent your existing SSIS packages to run on Azure-SSIS Integration Runtime environments.

- Information issues: SSIS features within your packages that are only partially supported, or are deprecated. Regardless of which category of information you receive, the Data Migration Assistant will perform the assessment on a batch of SSIS packages and provide guidance and potential mitigation steps that you can use to address the blockers and issues that are raised.

Perform a migration of your packages

Before migrating, you must know which Azure SQL database service you are migrating to. This can include migrating to Azure SQL Managed Instance (MI), or Azure SQL Database. Furthermore, when migrating SSIS packages. you have to consider the location of the SSIS packages that you are migrating, as this can impact how you migrate the packages, and which tool you will need to use. There are four types of storage including:

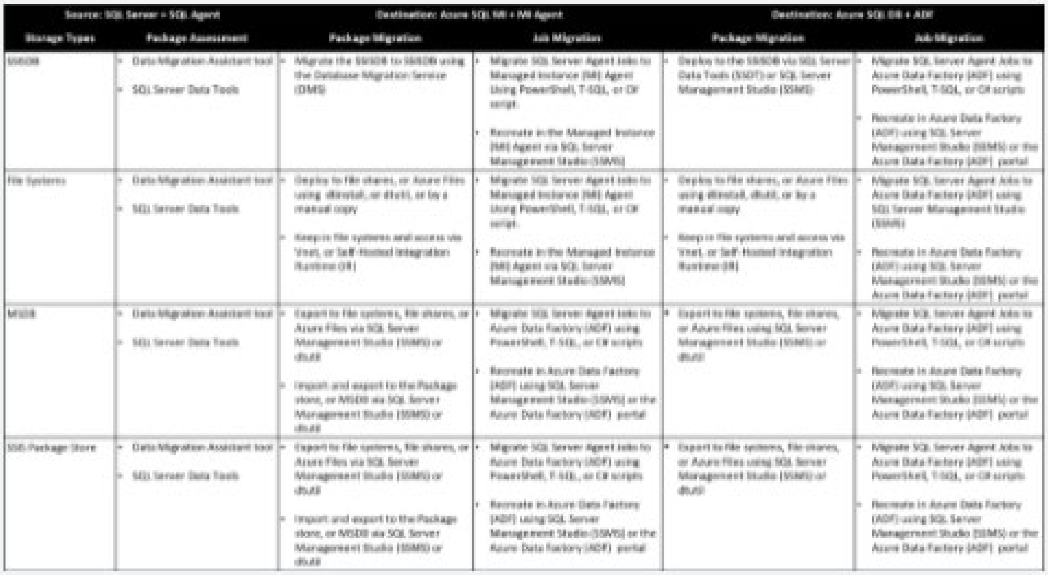
- SSIS Catalog (also known as SSISDB)

- File System

- MSDB database in SQL Server

- SSIS Package store

Based on this information, you can use the following table as a basis for understanding the tools you can use to perform migration assessments, and to perform the migration itself.



Source: SQL Server + SQL Agent		Destination: Azure SQL MI + MI Agent		Destination: Azure SQL DB + ADF	
Storage Types	Package Assessment	Package Migration	Job Migration	Package Migration	Job Migration
SSISDB	• Data Migration Assistant tool • SQL Server Data Tools	• Migrate the SSISDB to SSISDB using the Database Migration Service (DMS)	• Migrate SQL Server Agent Jobs to Managed Instance (MI) Agent Using PowerShell, T-SQL, or C# script. • Recreate in the Managed Instance (MI) Agent via SQL Server Management Studio (SSMS)	• Deploy to the SSISDB via SQL Server Data Tools (SSDT) or SQL Server Management Studio (SSMS)	• Migrate SQL Server Agent Jobs to Azure Data Factory (ADF) using PowerShell, T-SQL, or C# scripts • Recreate in Azure Data Factory (ADF) using SQL Server Management Studio (SSMS) or the Azure Data Factory (ADF) portal
File Systems	• Data Migration Assistant tool • SQL Server Data Tools	• Deploy to file shares, or Azure Files using dtinstall, or dtutil, or by a manual copy • Keep in file systems and access via Vnet, or Self-Hosted Integration Runtime (IR)	• Migrate SQL Server Agent Jobs to Managed Instance (MI) Agent Using PowerShell, T-SQL, or C# script. • Recreate in the Managed Instance (MI) Agent via SQL Server Management Studio (SSMS)	• Deploy to file shares, or Azure Files using dtinstall, dtutil, or by a manual copy • Keep in file systems and access via Vnet, or Self-Hosted Integration Runtime (IR)	• Migrate SQL Server Agent Jobs to Azure Data Factory (ADF) using SQL Server Management Studio (SSMS) • Recreate in Azure Data Factory (ADF) using SQL Server Management Studio (SSMS) or the Azure Data Factory (ADF) portal
MSDB	• Data Migration Assistant tool • SQL Server Data Tools	• Export to file systems, file shares, or Azure Files via SQL Server Management Studio (SSMS) or dtutil • Import and export to the Package store, or MSDB via SQL Server Management Studio (SSMS) or dtutil	• Migrate SQL Server Agent Jobs to Azure Data Factory (ADF) using PowerShell, T-SQL, or C# scripts • Recreate in Azure Data Factory (ADF) using SQL Server Management Studio (SSMS) or the Azure Data Factory (ADF) portal	• Export to file systems, file shares, or Azure Files using SQL Server Management Studio (SSMS) or dtutil	• Migrate SQL Server Agent Jobs to Azure Data Factory (ADF) using PowerShell, T-SQL, or C# scripts • Recreate in Azure Data Factory (ADF) using SQL Server Management Studio (SSMS) or the Azure Data Factory (ADF) portal
SSIS Package Store	• Data Migration Assistant tool • SQL Server Data Tools	• Export to file systems, file shares, or Azure Files via SQL Server Management Studio (SSMS) or dtutil • Import and export to the Package store, or MSDB via SQL Server Management Studio (SSMS) or dtutil	• Migrate SQL Server Agent Jobs to Azure Data Factory (ADF) using PowerShell, T-SQL, or C# scripts • Recreate in Azure Data Factory (ADF) using SQL Server Management Studio (SSMS) or the Azure Data Factory (ADF) portal	• Export to file systems, file shares, or Azure Files using SQL Server Management Studio (SSMS) or dtutil	• Migrate SQL Server Agent Jobs to Azure Data Factory (ADF) using PowerShell, T-SQL, or C# scripts • Recreate in Azure Data Factory (ADF) using SQL Server Management Studio (SSMS) or the Azure Data Factory (ADF) portal

Microsoft Data Migration Assistant

The Data Migration Assistant 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 objects from your source server to your target server.

This tool can be helpful to you in identifying any issues that can affect a migration to an Azure SQL data platform. The DMA can run assessment projects that will identify any blocking issues or unsupported features that are currently in use with your on-premises SQL Server. It can also help you understand the new features in the target SQL Server platform that the database can benefit from after a migration. The DMA can also perform migration projects that can migrate an on-premises SQL Server instance to a modern SQL Server instance hosted on-premises or on an Azure virtual machine (VM) that is accessible from your on-premises network.

The Data Migration Assistant replaces all previous versions of SQL Server Upgrade Advisor and should be used for upgrades for most SQL Server versions.

https://www.sqlshack.com/move-local-ssis-packages-to-azure-data-factory/

Question 57: **Skipped**

What is the name of the application architecture that enables near real-time querying to provide insights?

- ○ OLAP
- ○ HTAP **(Correct)**
- ○ ELT
- ○ OLTP
- ○ ETL
- ○ ADPS

Explanation

HTAP stands for Hybrid Transactional and Analytical Processing that enable you to gain insights from operational systems without impacting the performance of the operational system.

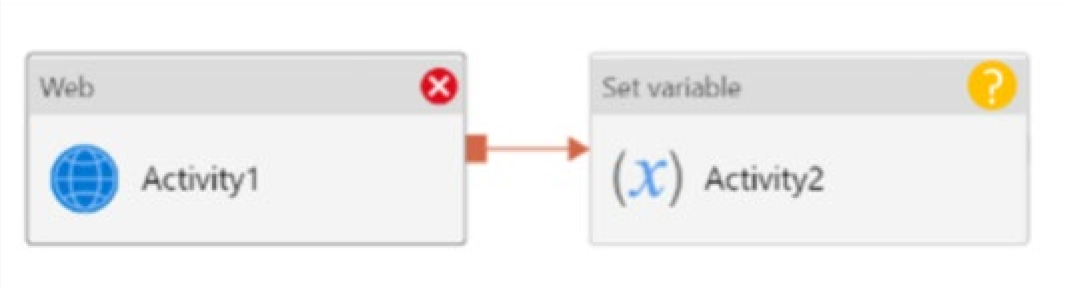
https://docs.microsoft.com/en-us/learn/paths/work-with-hybrid-transactional-analytical-processing-solutions/

https://docs.microsoft.com/en-us/learn/modules/design-hybrid-transactional-analytical-processing-using-azure-synapse-analytics/

https://www.zdnet.com/article/what-is-hybrid-transactionanalytical-processing-htap/

Question 58: Skipped

Scenario: We are working on a project which has a pipeline with two activities where Activity2 has a failure dependency on Activity1.

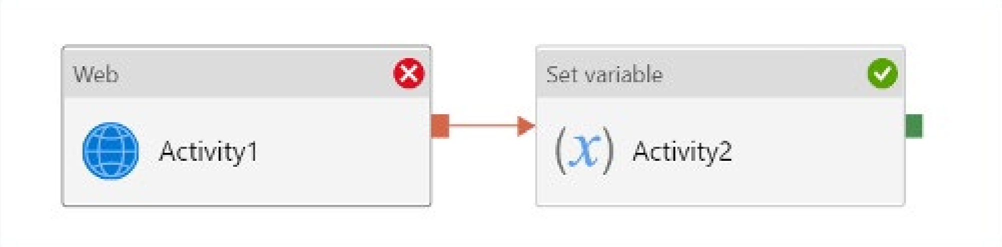


What will the result be of the pipeline?

- ○ This pipeline reports success. (Correct)
- ○ This pipeline reports failure.
- ○ This pipeline reports skipped.
- ○ This pipeline reports completed.

Explanation

If we have a pipeline with two activities where Activity2 has a failure dependency on Activity1, the pipeline will not fail just because Activity1 failed. If Activity1 fails and Activity2 succeeds, the pipeline will succeed. This scenario is treated as a try-catch block by Data Factory.



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

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

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

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

As a solution to the challenges with Data Lakes noted above, [?] is a file format that can help you build a data lake comprised of one or many tables in [?] format. [?] integrates tightly with Apache Spark, and uses an open format that is based on Parquet. Because it is an open-source format, [?] is also supported by other data platforms, including Azure Synapse Analytics.

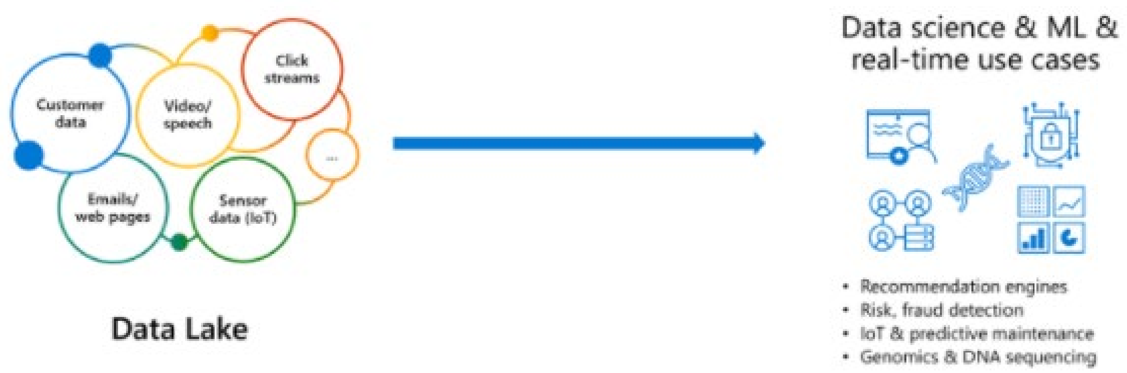
- ○ Augmenter
- ○ Data Organizer
- ○ Data Sea
- ○ Delta Lake (Correct)

Explanation
Delta Lake is a transactional storage layer designed specifically to work with Apache Spark and Databricks File System (DBFS). At the core of Delta Lake is an optimized Spark table. It stores your data as Apache Parquet files in DBFS and maintains a transaction log that efficiently tracks changes to the table.

Data lakes

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

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



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

The challenge with data lakes

Why are these projects struggling with reliability and performance?

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

- Schema enforcement when new tables are introduced.

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

- Frequent refreshes of metadata.

- Bottlenecks of small file sizes for distributed computations.

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

There are also data reliability challenges with data lakes:

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

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

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

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

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

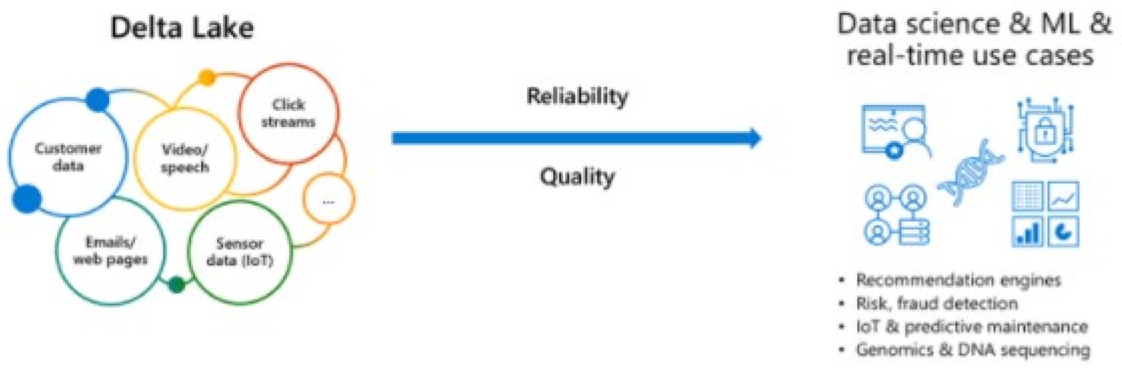
- **Partitioning also known as "poor man's indexing"**- breaks down if you picked the wrong fields or when data has many dimensions, high cardinality columns.

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

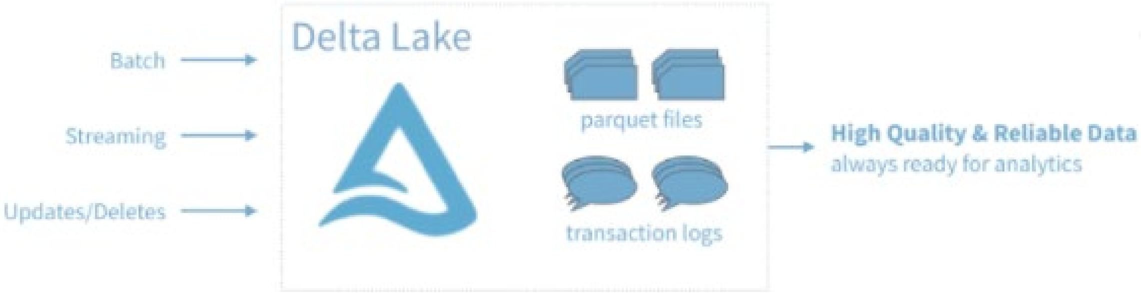
The solution: Delta Lake

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

Delta Lake makes data ready for analytics.



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



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

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

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

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

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

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

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

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

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

Get started with Delta using Spark APIs

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

Instead of parquet...

```
Python

CREATE TABLE ...

USING parquet

...


dataframe

.write

.format("parquet")

.save("/data")

... simply say delta

Python

CREATE TABLE ...

USING delta

...


dataframe

.write

.format("delta")
```

```
.save("/data")
```

Using Delta with your existing Parquet tables

Step 1: Convert Parquet to Delta tables:

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

Basic syntax

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

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

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

- `INSERT` a row

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

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

```
SQL
MERGE INTO customers -- Delta table
USING updates
ON customers.customerId = source.customerId
WHEN MATCHED THEN
UPDATE SET address = updates.address
```

```
WHEN NOT MATCHED

THEN INSERT (customerId, address) VALUES (updates.customerId, updates.address)
```

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

Other time travel use cases are:

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

• Writing complex temporal queries.

• Fixing mistakes in your data.

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

Example of using time travel to reproduce experiments and reports:

```
SQL

SELECT count(*) FROM events

TIMESTAMP AS OF timestamp



SELECT count(*) FROM events

VERSION AS OF version

Python

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

If you need to rollback accidental or bad writes:

SQL

INSERT INTO my_table

SELECT * FROM my_table TIMESTAMP AS OF

date_sub( current_date(), 1)
```

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

Question 60: Skipped

What is a dataframe?

- ○ A creation of a data structure (Correct)
- ○ A CSV file
- ○ An Array
- ○ A parquet file

Explanation

A DataFrame creates a data structure and it's one of the core data structures in Spark.

What are dataframes?

Basically you could view DataFrames as you might see in excel. It's like a box with squares in it, that organizes data, which we could also refer to as a table of data.

What does a table of data mean?

It is a single set of two-dimensional data that can have multiple rows and columns in the data. Each row, is a sample of data. Each column is a variable or parameter that is able to describe the row that contains the sample of data.

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.

What you see in Data Engineering is that you start with reading or loading data that can be unstructured, semi-structured, or structured, which is stored in a DataFrame and start transforming that data in order to get insights. You can use different functionalities in order to do so, like using Spark SQL, PySpark, and others.

Usually when you see 'df' in some code it refers to a dataframe.

You can either create your own dataframe as this example shows:

```
Python

new_rows = [('CA',22, 45000),("WA",35,65000) ,("WA",50,85000)]
```

```
demo_df = spark.createDataFrame(new_rows, ['state', 'age', 'salary'])

demo_df.show()
```

Or load a file that contains data into a dataframe like in the below example where the open taxi dataset is used:

```
Python

from azureml.opendatasets import NycTlcYellow



data = NycTlcYellow()

data_df = data.to_spark_dataframe()

display(data_df.limit(10))
```

Once you're at the stage where you'd like to manipulate the data that is stored in a DataFrame, you can use User-Defined Functions (UDFs) that are column-based and help you transform and manipulate the data stored in a DataFrame.

https://www.tutorialspoint.com/spark_sql/spark_sql_dataframes.htm

Question 61: **Skipped**

Scenario: Queen Consolidated was overtaken by Raymond Carson Palmer and rebranded as Palmer Technologies. Now that Ray is overseeing the operations at Palmer, Ray has decided to implement on-premises Microsoft SQL Server pipelines by using a custom solution.

Currently, you are in a meeting with the IT team and discussing a project to pull data from SQL Server and migrate it to Azure Blob storage.

Required:

• The process must orchestrate and manage the data lifecycle.

• The process must configure Azure Data Factory to connect to the on-premises SQL Server database.

Ray and the IT team have put together a list of actions they think need to be performed to meet the needs of the project, but they are not sure on the order to execute. Below is a list of the actions they are considering.

Proposed Actions:

a. Create an Azure Data Factory resource.

b. Configure a self-hosted integration runtime.

c. Create a virtual private network (VPN) connection from on-prem to MS Azure.

d. Create a database master key on SQL Server.

e. Backup the database and send it to Azure Blob storage.

f. Configure the on-prem SQL Server instance with an integration runtime.

As you are the Azure SME, Ray and the team look to you for direction on selecting the required items and putting them in the proper order. Which of the below contains the correct items in the correct sequence to meet the requirements?

- ○
  a → b → f
  (Correct)
- ○
  c → a → b → f

- ○
  c → d → a → b → f
- ○
  d → c → e → b

Explanation
Step 1: Create an Azure Data Factory

The instructions for creating a new Azure Data Factory and a resource group in the Azure portal are provided Create an Azure Data Factory. Name the new ADF instance adfdsp and name the resource group created adfdsprg.

Step 2: Install and configure Azure Data Factory Integration Runtime

The Integration Runtime is a customer-managed data integration infrastructure used by Azure Data Factory to provide data integration capabilities across different network environments. This runtime was formerly called "Data Management Gateway".

To set up, follow the instructions for creating a pipeline

Step 3: Configure the on-prem SQL Server instance with an integration runtime.

Create linked services to connect to the data resources. A linked service defines the information needed for Azure Data Factory to connect to a data resource. We have three resources in this scenario for which linked services are needed:

1. On-premises SQL Server

2. Azure Blob Storage

3. Azure SQL Database

https://docs.microsoft.com/pt-pt/azure/machine-learning/team-data-science-process/move-sql-azure-adf

It's not necessary to *Create a virtual private network (VPN) connection from on-premises to Microsoft Azure* - all communication from IR to ADF is over HTTPS, ∴ **VPN is not a Required item.**

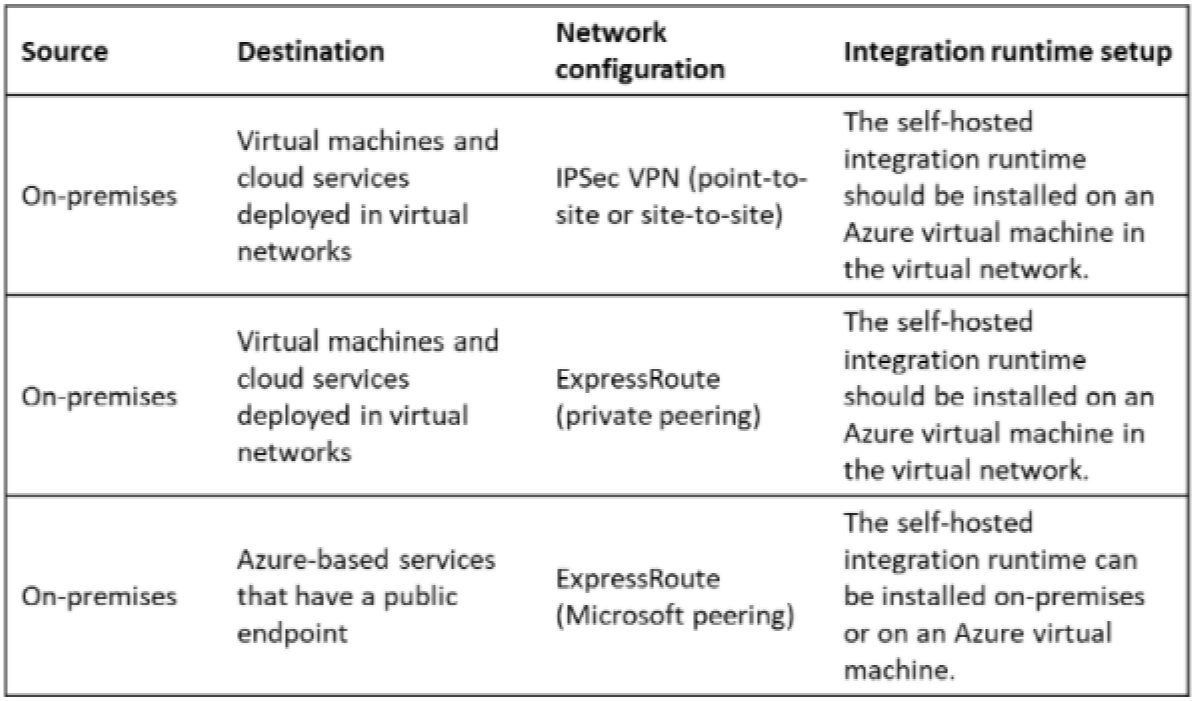
Encryption in transit

All data transfers are via secure channel HTTPS and TLS over TCP to prevent man-in-the-middle attacks during communication with Azure services.

You can also use IPSec VPN or Azure ExpressRoute to further secure the communication channel between your on-premises network and Azure.

Azure Virtual Network is a logical representation of your network in the cloud. You can connect an on-premises network to your virtual network by setting up IPSec VPN (site-to-site) or ExpressRoute (private peering).

The following table summarizes the network and self-hosted integration runtime configuration recommendations based on different combinations of source and destination locations for hybrid data movement.



Source	Destination	Network configuration	Integration runtime setup
On-premises	Virtual machines and cloud services deployed in virtual networks	IPSec VPN (point-to-site or site-to-site)	The self-hosted integration runtime should be installed on an Azure virtual machine in the virtual network.
On-premises	Virtual machines and cloud services deployed in virtual networks	ExpressRoute (private peering)	The self-hosted integration runtime should be installed on an Azure virtual machine in the virtual network.
On-premises	Azure-based services that have a public endpoint	ExpressRoute (Microsoft peering)	The self-hosted integration runtime can be installed on-premises or on an Azure virtual machine.

https://docs.microsoft.com/en-us/azure/data-factory/data-movement-security-considerations

Move data from a SQL Server database to the SQL Database with Azure Data Factory

AZure Data Factory is a fully managed cloud-based data integration service that orchestrates and automates the movement and transformation of data. The key concept in the ADF model is the pipeline. A pipeline is a logical grouping of Activities, each of which defines the actions to be performed on the data contained in the Data Sets. The linked services are used to define the information necessary for the Data Factory to connect to the data resources.

With the ADF, existing data processing services can be composed of highly available data pipelines and managed in the cloud. These data pipelines can be programmed to ingest, prepare, transform, analyze and publish data, and the ADF manages and orchestrates the complex data and processing dependencies. Solutions can be quickly built and deployed in the cloud, connecting an increasing number of data sources on-premises and in the cloud.

Consider using the ADF:

• when data needs to be continuously migrated in a hybrid scenario that accesses both on-premises and cloud resources

• when data needs transformation or has business logic added to it when it is migrated.

The ADF allows scheduling and monitoring of jobs using simple JSON scripts that manage the movement of data on a periodic basis. ADF also has other capabilities, such as supporting complex operations. For more information about the ADF, see the documentation at Azure Data Factory (ADF).

The set

We created an ADF pipeline that comprises two data migration activities. Together, they move data daily between a SQL Server database and the Azure SQL Database. The two activities are:

• Copy data from a SQL Server database to an Azure Blob storage account.

• Copy the data from the Azure Blob storage account to the Azure SQL Database.

Upload the data to your instance of SQL Server

Create an Azure Data Factory

Instructions for creating a new Azure Data Factory and resource group on the Azure portal are provided Create an Azure Data Factory. Name the new adf instance adfdsp and name the resource group created adfdsprg.

Install and configure Azure data factory integration time

Integration Runtime is a customer-managed data integration infrastructure used by Azure Data Factory to provide data integration capabilities in different network environments. This uptime was previously called "Data Management Gateway".

https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/move-sql-azure-adf

Question 62: Skipped

Scenario: Big Belly Foods, Inc. (BB) owns and operates 300 convenience stores across LatAm. The company sells a variety of packaged foods and drinks, as well as a variety of prepared foods, such as sandwiches and pizzas. The company has a loyalty club whereby members can get daily discounts on specific items by providing their membership number at checkout.

BB employs business analysts who prefer to analyze data by using Microsoft Power BI, and data scientists who prefer analyzing data in Azure Databricks notebooks. You have been hired as an Azure Expert SME and you are to consult the IT team on various Azure related projects.

Business Requirements:

BB wants to create a new analytics environment in Azure to meet the following requirements:

• See inventory levels across the stores. Data must be updated as close to real time as possible.

• Execute ad hoc analytical queries on historical data to identify whether the loyalty club discounts increase sales of the discounted products.

• Every four hours, notify store employees about how many prepared food items to produce based on historical demand from the sales data.

Technical Requirements:

BB identifies the following technical requirements:

• Minimize the number of different Azure services needed to achieve the business goals.

• Use platform as a service (PaaS) offerings whenever possible and avoid having to provision virtual machines that must be managed by BB.

• Ensure that the analytical data store is accessible only to the company's on-premises network and Azure services.

• Use Azure Active Directory (Azure AD) authentication whenever possible.

• Use the principle of least privilege when designing security.

• Stage Inventory data in Azure Data Lake Storage Gen2 before loading the data into the analytical data store. BB wants to remove transient data from Data

- Lake Storage once the data is no longer in use. Files that have a modified date that is older than 14 days must be removed.

- Limit the business analysts' access to customer contact information, such as phone numbers, because this type of data is not analytically relevant.

- Ensure that you can quickly restore a copy of the analytical data store within one hour in the event of corruption or accidental deletion.

Planned Environment:

BB plans to implement the following environment:

- The application development team will create an Azure event hub to receive real-time sales data, including store number, date, time, product ID, customer loyalty number, price, and discount amount, from the point of sale (POS) system and output the data to data storage in Azure.

- Customer data, including name, contact information, and loyalty number, comes from Salesforce, a SaaS application, and can be imported into Azure once every eight hours. Row modified dates are not trusted in the source table.

- Product data, including product ID, name, and category, comes from Salesforce and can be imported into Azure once every eight hours. Row modified dates are not trusted in the source table.

- Daily inventory data comes from a Microsoft SQL server located on a private network.

- BB currently has 5 TB of historical sales data and 100 GB of customer data. The company expects approximately 100 GB of new data per month for the next year.

- BB will build a custom application named FoodPrep to provide store employees with the calculation results of how many prepared food items to produce every four hours.

- BB does not plan to implement Azure ExpressRoute or a VPN between the on-premises network and Azure.

The Ask:

The team looks to you for direction on what should be used to import the daily inventory data from the SQL server to Azure Data Lake Storage. Which Azure Data Factory components should you recommend for the trigger type?

- ○ Tumbling window trigger
- ○ Scaling window trigger
- ○ Schedule trigger
  (Correct)
- ○ Event-based trigger

Explanation
The following are the recommends you should present:

• A self-hosted IR is capable of running copy activity between a cloud data stores and a data store in private network.

• Schedule trigger set for an 8 hour interval.

• A copy activity type

Rational:

• Customer data, including name, contact information, and loyalty number, comes from Salesforce and can be imported into Azure once every eight hours. Row modified dates are not trusted in the source table.

• Product data, including product ID, name, and category, comes from Salesforce and can be imported into Azure once every eight hours. Row modified dates are not trusted in the source table.

Create a trigger that runs a pipeline on a schedule

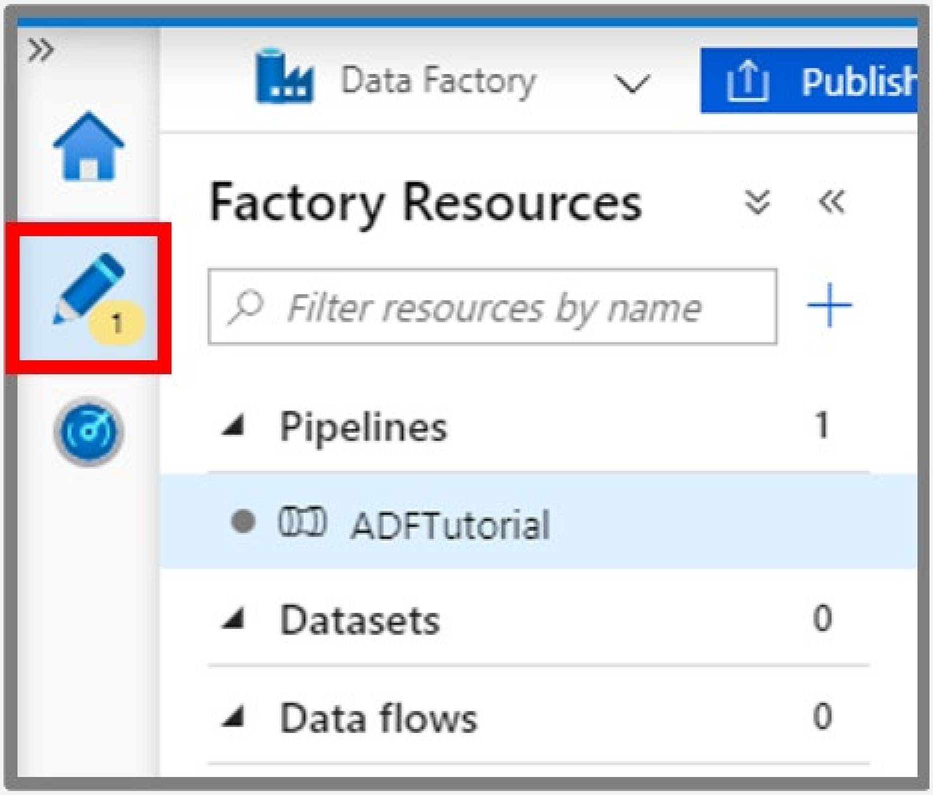
When creating a schedule trigger, you specify a schedule (start date, recurrence, end date etc.) for the trigger, and associate with a pipeline. Pipelines and triggers have a many-to-many relationship. Multiple triggers can kick off a single pipeline. A single trigger can kick off multiple pipelines.

Note: For a complete walkthrough of creating a pipeline and a schedule trigger, which associates the trigger with the pipeline, and runs and monitors the pipeline, see Quickstart: create a data factory using Data Factory UI.

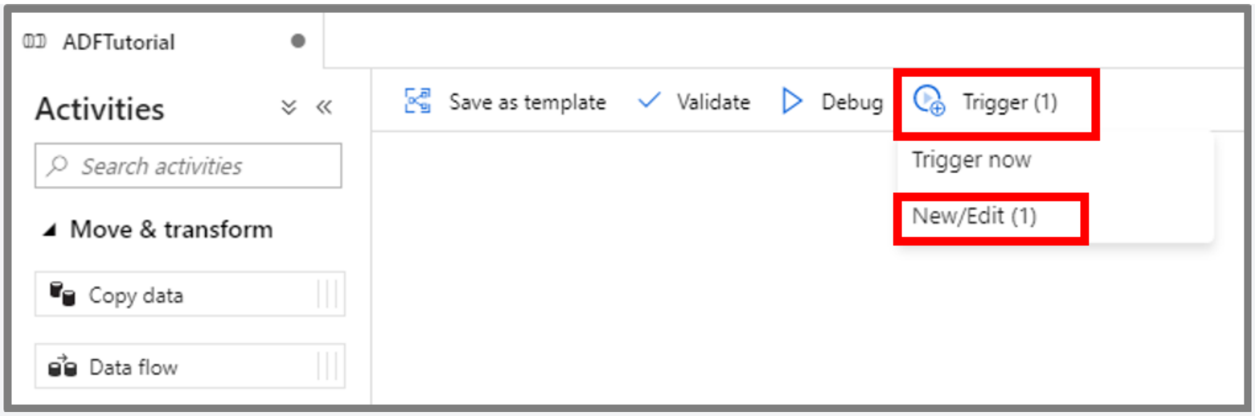
Data Factory UI

You can create a **schedule trigger** to schedule a pipeline to run periodically (hourly, daily, etc.).

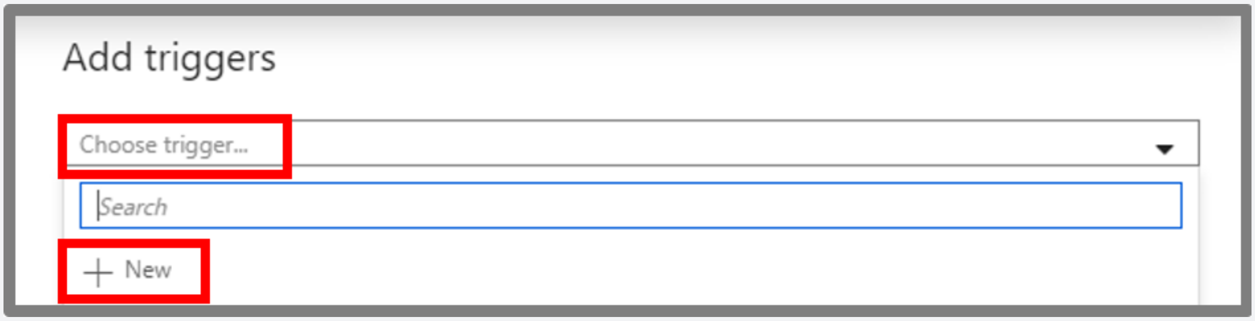
1. Switch to the **Edit** tab, shown with a pencil symbol.



2. Select **Trigger** on the menu, then select **New/Edit**.



3. On the **Add Triggers** page, select **Choose trigger...**, then select **+New**.



4. On the **New Trigger** page, do the following steps:

• Confirm that **Schedule** is selected for **Type**.

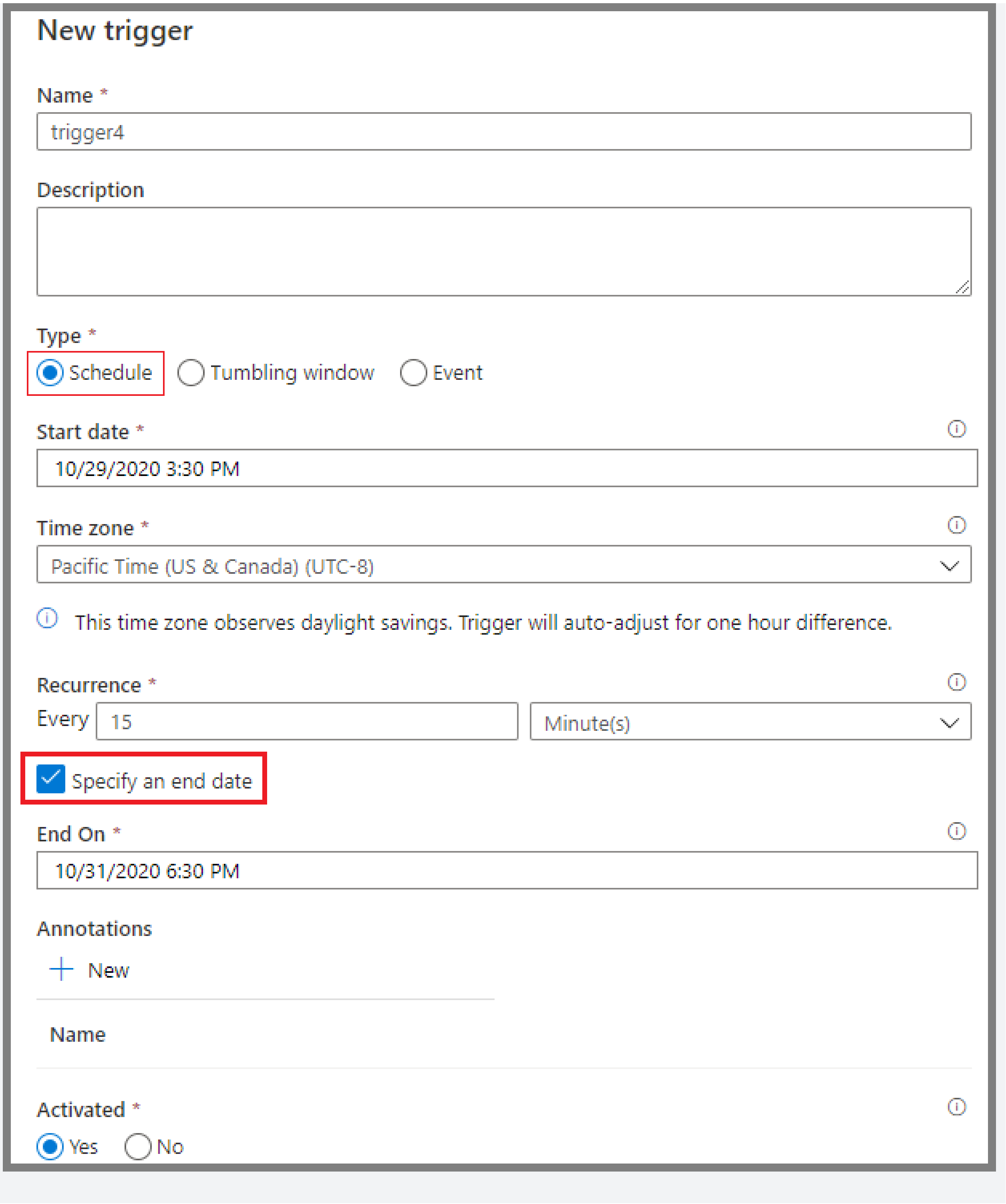
• Specify the start datetime of the trigger for **Start Date**. It's set to the current datetime in Coordinated Universal Time (UTC) by default.

• Specify the time zone that the trigger will be created in. The time zone setting will apply to **Start Date**, **End Date**, and **Schedule Execution Times** in Advanced recurrence options. Changing Time Zone setting will not automatically change your start date. Make sure the Start Date is correct in the specified time zone. Please note that Scheduled Execution time of Trigger will be considered post the Start Date (Ensure Start Date is atleast 1minute lesser than the Execution time else it will trigger pipeline in next recurrence).

Note: For time zones that observe daylight saving, trigger time will auto-adjust for the twice a year change. To opt out of the daylight saving change, please select a time zone that does not observe daylight saving, for instance UTC

• Specify **Recurrence** for the trigger. Select one of the values from the drop-down list (Every minute, Hourly, Daily, Weekly, and Monthly). Enter the multiplier in the text box. For example, if you want the trigger to run once for every 15 minutes, you select **Every Minute**, and enter **15** in the text box.

• To specify an end date time, select **Specify an End Date**, and specify *Ends On*, then select **OK**. There is a cost associated with each pipeline run. If you are testing, you may want to ensure that the pipeline is triggered only a couple of times. However, ensure that there is enough time for the pipeline to run between the publish time and the end time. The trigger comes into effect only after you publish the solution to Data Factory, not when you save the trigger in the UI.



New trigger

Name *

trigger4

Description

Type *

● Schedule ○ Tumbling window ○ Event

Start date *

10/29/2020 3:30 PM

Time zone *

Pacific Time (US & Canada) (UTC-8)

This time zone observes daylight savings. Trigger will auto-adjust for one hour difference.

Recurrence *

Every 15 Minute(s)

☑ Specify an end date

End On *

10/31/2020 6:30 PM

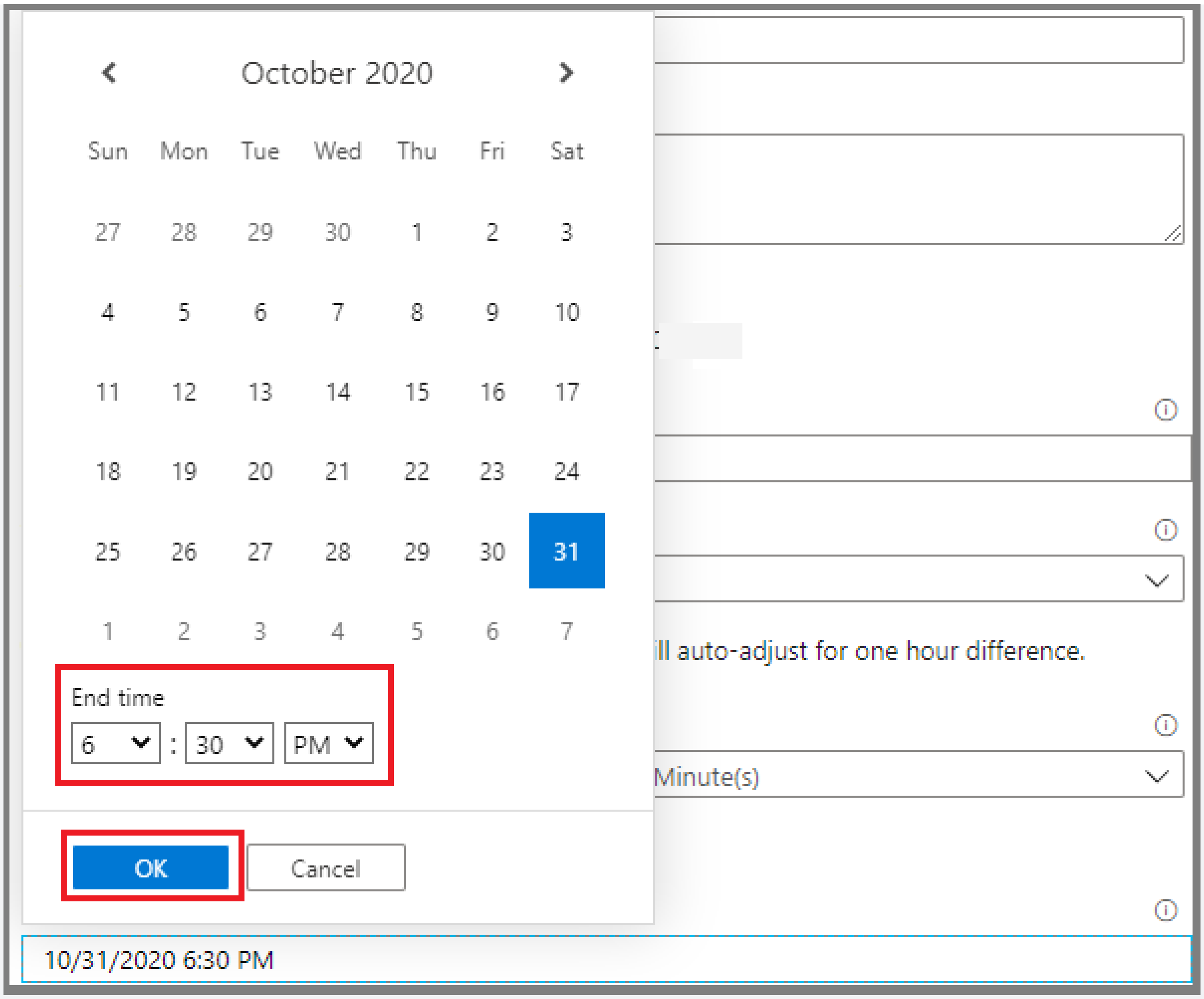
Annotations

+ New

Name

Activated *

● Yes ○ No



October 2020

Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

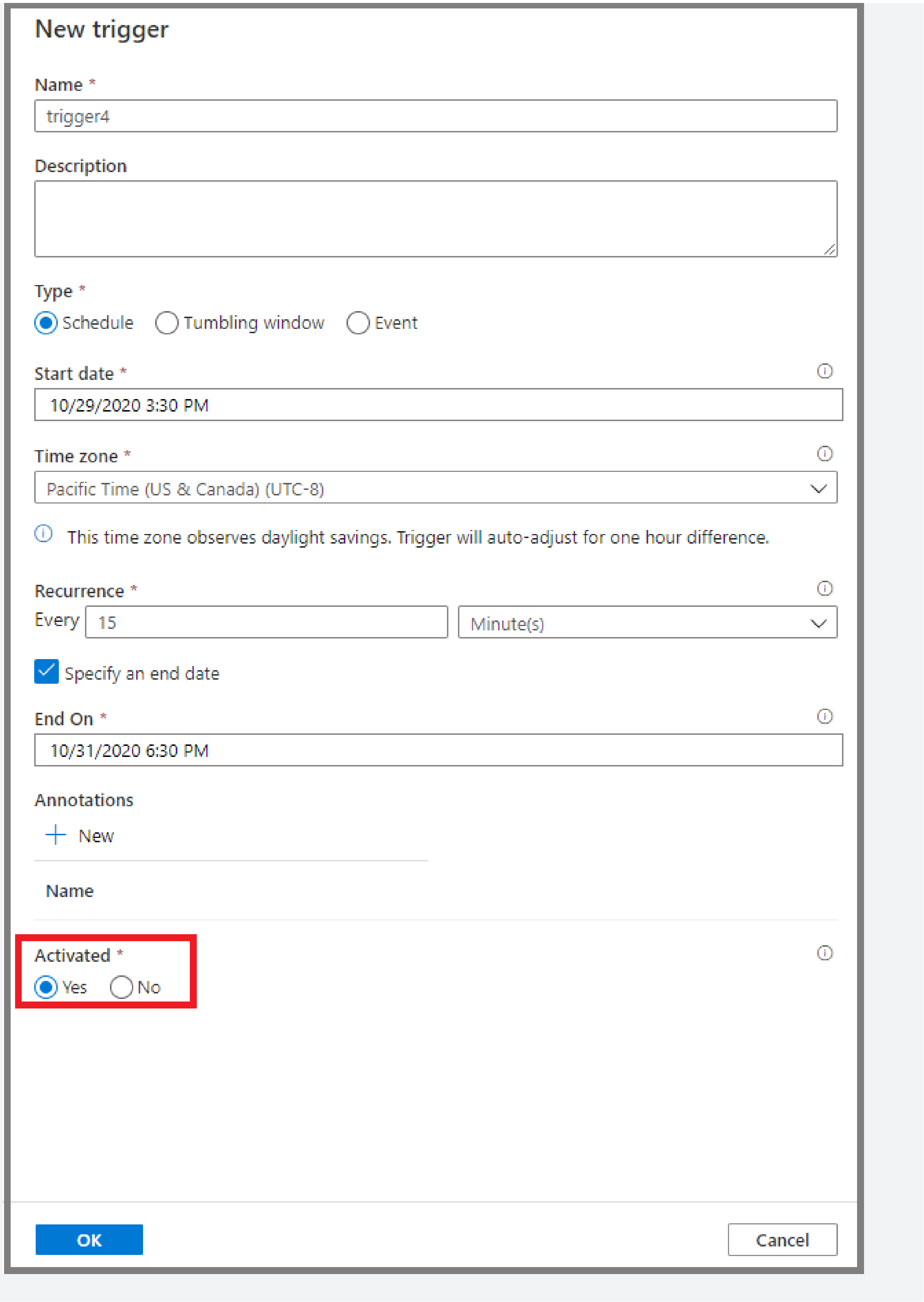
End time 6 : 30 PM

ll auto-adjust for one hour difference.

Minute(s)

10/31/2020 6:30 PM

5. In the **New Trigger** window, select **Yes** in the **Activated** option, then select **OK**. You can use this checkbox to deactivate the trigger later.



New trigger

Name *

trigger4

Description

Type *

● Schedule ○ Tumbling window ○ Event

Start date *

10/29/2020 3:30 PM

Time zone *

Pacific Time (US & Canada) (UTC-8)

This time zone observes daylight savings. Trigger will auto-adjust for one hour difference.

Recurrence *

Every 15 Minute(s)

☑ Specify an end date

End On *

10/31/2020 6:30 PM

Annotations

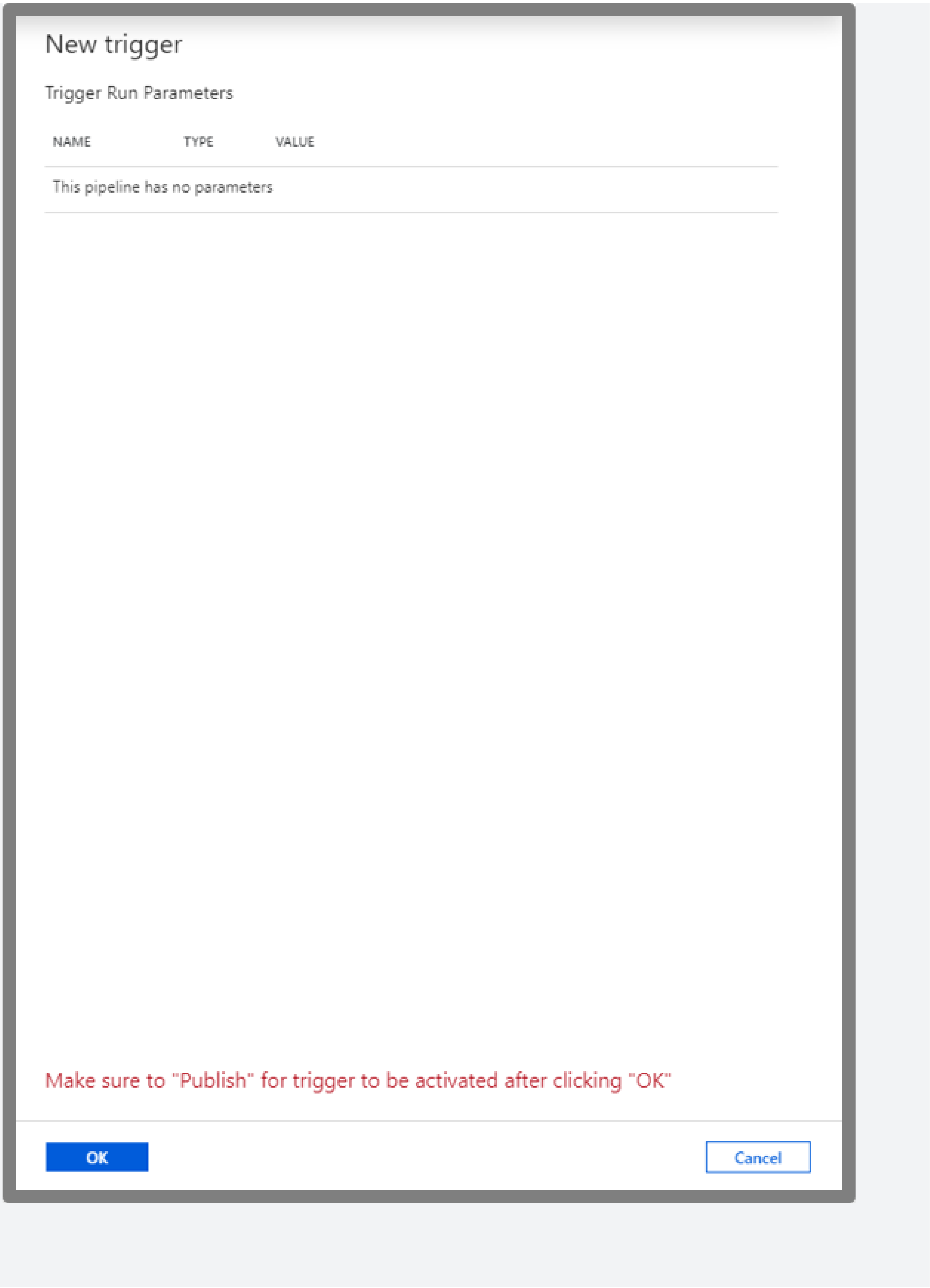
+ New

Name

Activated *

● Yes ○ No

6. In the **New Trigger** window, review the warning message, then select **OK**.



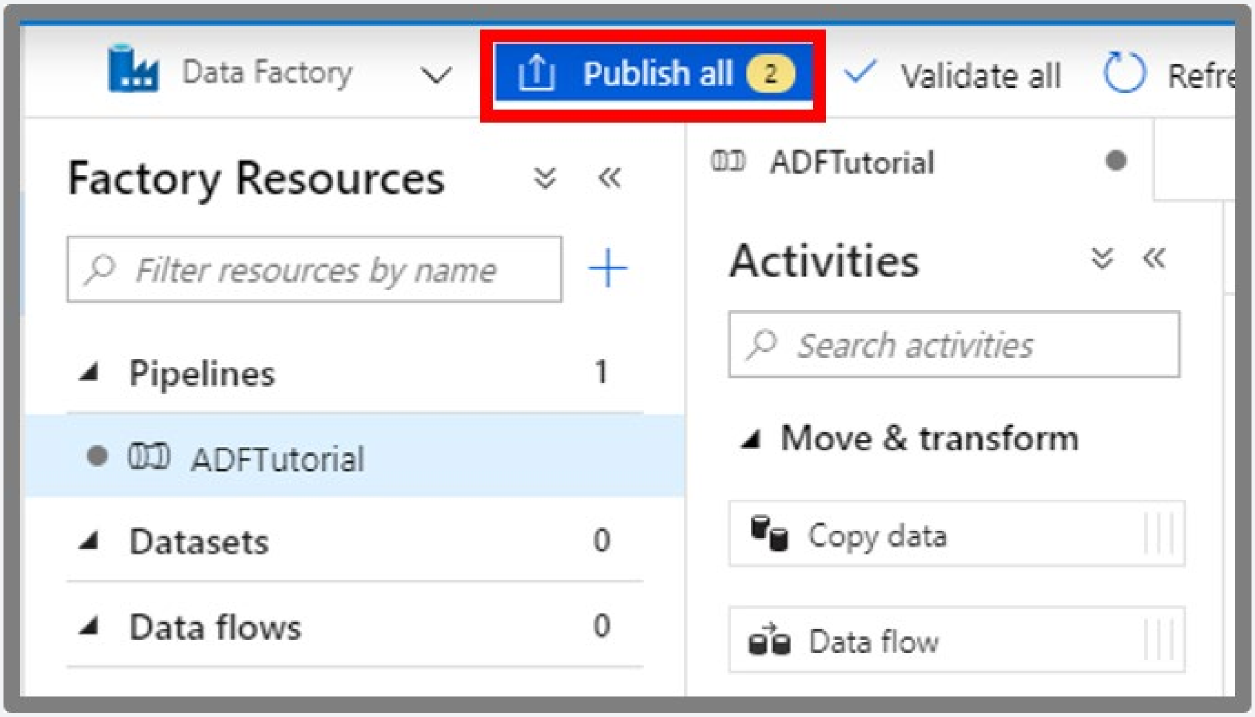
New trigger

Trigger Run Parameters

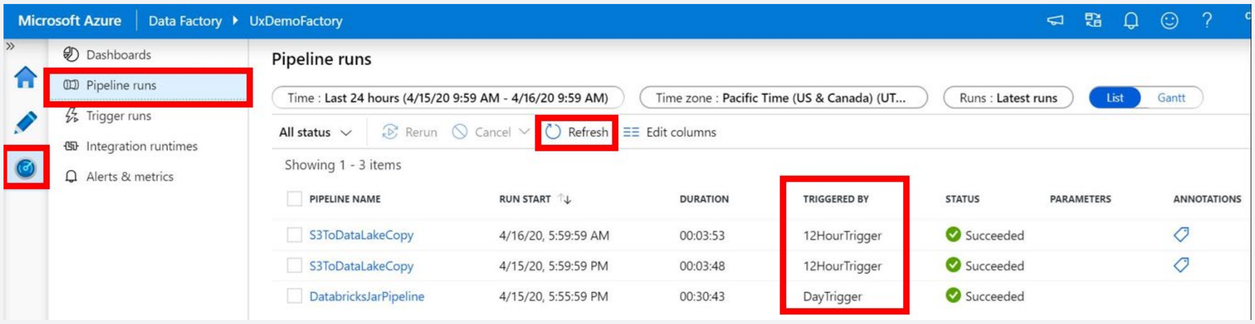
NAME	TYPE	VALUE
This pipeline has no parameters		

Make sure to "Publish" for trigger to be activated after clicking "OK"

7. Select **Publish all** to publish the changes to Data Factory. Until you publish the changes to Data Factory, the trigger doesn't start triggering the pipeline runs.



8. Switch to the **Pipeline runs** tab on the left, then select **Refresh** to refresh the list. You will see the pipeline runs triggered by the scheduled trigger. Notice the values in the **Triggered By** column. If you use the **Trigger Now** option, you will see the manual trigger run in the list.



Pipeline runs

Time : Last 24 hours (4/15/20 9:59 AM - 4/16/20 9:59 AM)

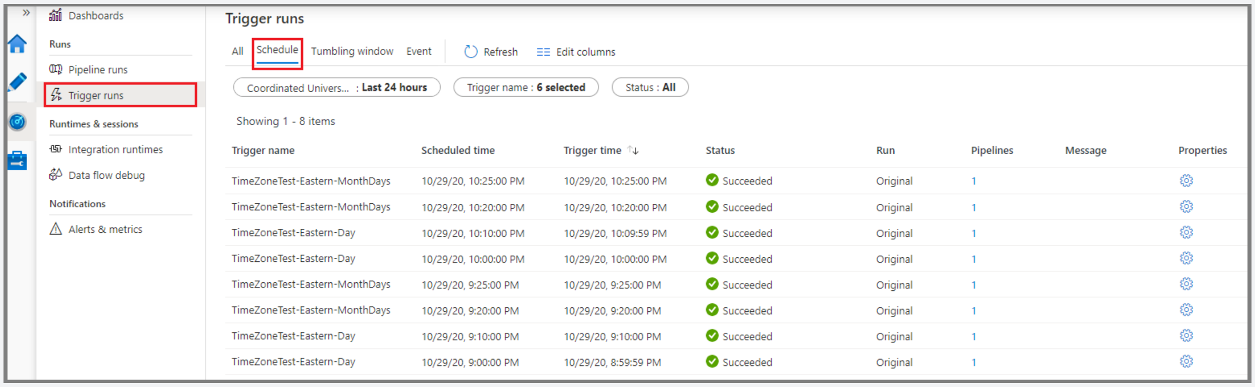
Time zone : Pacific Time (US & Canada) (UT...

Runs : Latest runs

Showing 1 - 3 items

PIPELINE NAME	RUN START	DURATION	TRIGGERED BY	STATUS	PARAMETERS	ANNOTATIONS
S3ToDataLakeCopy	4/16/20, 5:59:59 AM	00:03:53	12HourTrigger	✓ Succeeded		
S3ToDataLakeCopy	4/15/20, 5:59:59 PM	00:03:48	12HourTrigger	✓ Succeeded		
DatabricksJarPipeline	4/15/20, 5:55:59 PM	00:30:43	DayTrigger	✓ Succeeded		

9. Switch to the **Trigger Runs \ Schedule** view.



Coordinated Univers... : **Last 24 hours** | Trigger name : **6 selected** | Status : **All**

Showing 1 - 8 items

Trigger name	Scheduled time	Trigger time ↑↓	Status	Run	Pipelines	Message	Properties
TimeZoneTest-Eastern-MonthDays	10/29/20, 10:25:00 PM	10/29/20, 10:25:00 PM	✓ Succeeded	Original	1		⚙
TimeZoneTest-Eastern-MonthDays	10/29/20, 10:20:00 PM	10/29/20, 10:20:00 PM	✓ Succeeded	Original	1		⚙
TimeZoneTest-Eastern-Day	10/29/20, 10:10:00 PM	10/29/20, 10:09:59 PM	✓ Succeeded	Original	1		⚙
TimeZoneTest-Eastern-Day	10/29/20, 10:00:00 PM	10/29/20, 10:00:00 PM	✓ Succeeded	Original	1		⚙
TimeZoneTest-Eastern-MonthDays	10/29/20, 9:25:00 PM	10/29/20, 9:25:00 PM	✓ Succeeded	Original	1		⚙
TimeZoneTest-Eastern-MonthDays	10/29/20, 9:20:00 PM	10/29/20, 9:20:00 PM	✓ Succeeded	Original	1		⚙
TimeZoneTest-Eastern-Day	10/29/20, 9:10:00 PM	10/29/20, 9:10:00 PM	✓ Succeeded	Original	1		⚙
TimeZoneTest-Eastern-Day	10/29/20, 9:00:00 PM	10/29/20, 8:59:59 PM	✓ Succeeded	Original	1		⚙

https://docs.microsoft.com/en-us/azure/data-factory/how-to-create-schedule-trigger

Question 63: Skipped

See the following code:

```
PowerShell
$SubscriptionId = "add your subscription here"

Add-AzureRmAccount
Set-AzureRmContext -SubscriptionId $SubscriptionId

Register-AzureRmResourceProvider -ProviderNamespace Microsoft.DataFactory

$resourceGroupName = "cto_ignite"
$rglocation = "West US 2"

New-AzureRmDataFactoryV2 -ResourceGroupName $resourceGroupName -Name "ctoigniteADF" -Location $rglocation
```

What is this code template used to setup?

- ○ Azure Synapse Spark
- ○ Azure SQL Datawarehouse
- ○ Azure Network Security Groups
- ○ Azure Storage Account
- ○ Azure Private Endpoint
- ○ Azure Data Factory **(Correct)**
- ○ Azure Linked Service

Explanation

It is easy to set up Azure Data Factory from within the Azure portal, you only require the following information:

• **Name**: The name of the Azure Data Factory instance

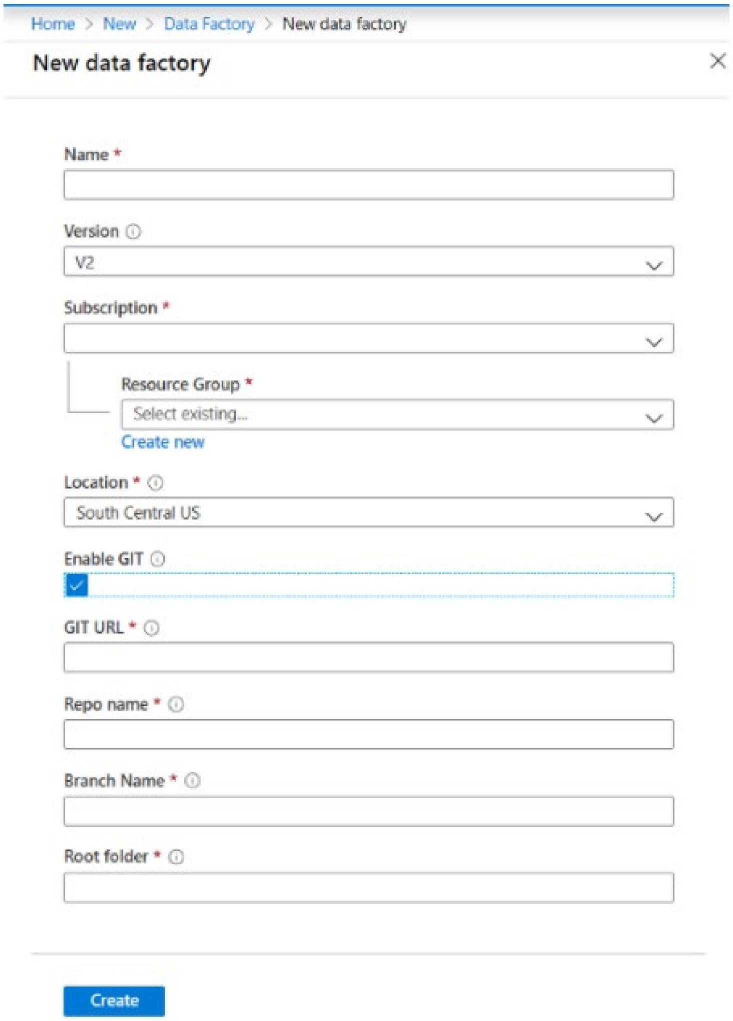
• **Subscription**: The subscription in which the ADF instance is created

• **Resource group**: The resource group where the ADF instance will reside

• **Version**: select V2 for the latest features

• **Location**: The datacentre location in which the instance is stored

Enable Git provides the capability to integrate the code that you create with a Git repository enabling you to source control the code that you would create. Define the GIT url, repository name, branch name, and the root folder.



Home > New > Data Factory > New data factory

New data factory

Name *

Version: V2

Subscription *

Resource Group *: Select existing...

Create new

Location *: South Central US

Enable GIT ☑

GIT URL *

Repo name *

Branch Name *

Root folder *

Create

Alternatively, there are a number of different ways that you can provision the service programmatically. In this example you can see PowerShell at work to set up the environment.

```
PowerShell
######################################################################
## PART I: Creating an Azure Data Factory ##
######################################################################



# Sign in to Azure and set the WINDOWS AZURE subscription to work with
$SubscriptionId = "add your subscription in the quotes"

Add-AzureRmAccount
Set-AzureRmContext -SubscriptionId $SubscriptionId

# register the Microsoft Azure Data Factory resource provider
Register-AzureRmResourceProvider -ProviderNamespace Microsoft.DataFactory

# DEFINE RESOURCE GROUP NAME AND LOCATION PARAMETERS
$resourceGroupName = "cto_ignite"
$rglocation = "West US 2"

# CREATE AZURE DATA FACTORY
New-AzureRmDataFactoryV2 -ResourceGroupName $resourceGroupName -Name "ctoigniteADF" -Location $rglocation
```

https://docs.microsoft.com/en-us/azure/data-factory/quickstart-create-data-factory-portal

Question 64: **Skipped**

Which Azure data platform is commonly used to process data in an ELT framework?

- ○ Azure Databricks
- ○ Azure Stream Analytics
- ○ Azure Data Factory **(Correct)**
- ○ Azure Data Catalog
- ○ Azure Data Lake Storage

Explanation

Azure Data Factory

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

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

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

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

Question 65: **Skipped**

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

Which of the following is best described by:

“By placing the init script in `/databricks/init folder`, you force the script’s execution every time any cluster is created or restarted by users of the workspace.”

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

Explanation

Favour cluster scoped init scripts over global and named scripts

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

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

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

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

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

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

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

Use cluster log delivery feature to manage logs

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

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

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

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

Question 66: **Skipped**

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

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

Which of the available roles is best described by:

“Works with artificial intelligence 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, they apply the pre-built capabilities of Cognitive Services APIs. Part of their job is to embed these capabilities within a new or existing application or bot. They rely on the expertise of data engineers to store information that's generated from artificial intelligence.”

- ○ Data Engineer
- ○ System Administrators
- ○ BI Engineer
- ○ AI Engineer **(Correct)**
- ○ Solution Architects
- ○ Project Manager
- ○ Data Scientist
- ○ RPA Developers

Explanation
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 pre-built 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 67: Skipped

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

Azure provides many ways to store your data. A storage account is a(n) [?] that groups a set of Azure Storage services together.

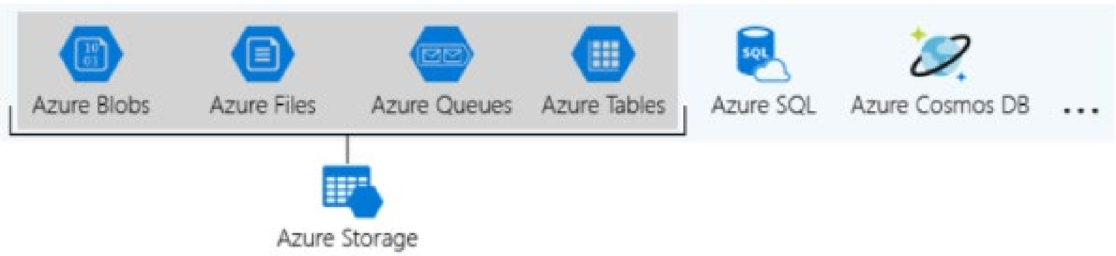
- ○ Structured dataset
- ○ Container (Correct)
- ○ Blob
- ○ Unstructured dataset
- ○ VM

Explanation

What is Azure Storage?

Azure provides many ways to store your data. There are multiple database options like Azure SQL Database, Azure Cosmos DB, and Azure Table Storage. Azure offers multiple ways to store and send messages, such as Azure Queues and Event Hubs. You can even store loose files using services like Azure Files and Azure Blobs.

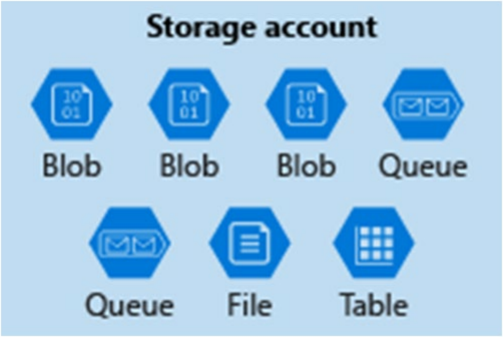
Azure selected four of these data services and placed them together under the name *Azure Storage*. The four services are Azure Blobs, Azure Files, Azure Queues, and Azure Tables. The following illustration shows the elements of Azure Storage.



These four were given special treatment because they are all primitive, cloud-based storage services and are often used together in the same application.

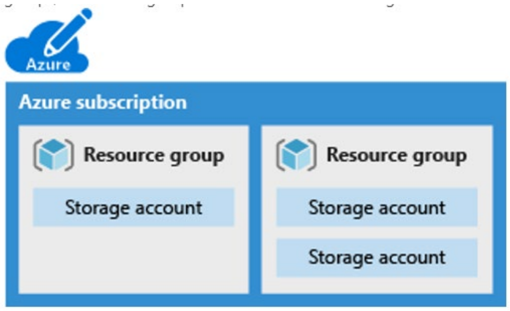
What is a storage account?

A *storage account* is a container that groups a set of Azure Storage services together. Only data services from Azure Storage can be included in a storage account (Azure Blobs, Azure Files, Azure Queues, and Azure Tables). The following illustration shows a storage account containing several data services.

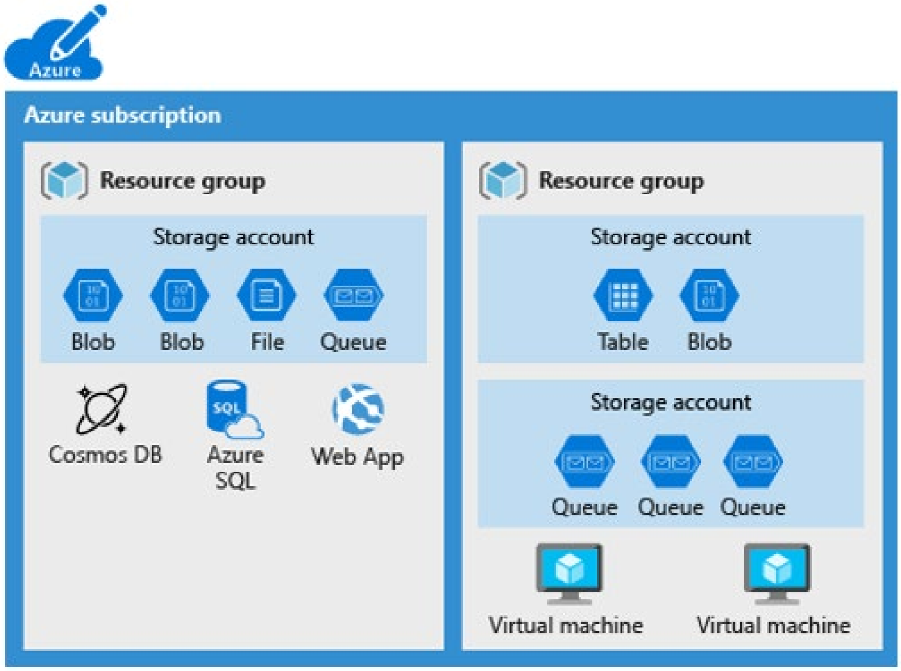


Combining data services into a storage account lets you manage them as a group. The settings you specify when you create the account, or any that you change after creation, are applied to everything in the account. Deleting the storage account deletes all of the data stored inside it.

A storage account is an Azure resource and is included in a resource group. The following illustration shows an Azure subscription containing multiple resource groups, where each group contains one or more storage accounts.



Other Azure data services like Azure SQL and Azure Cosmos DB are managed as independent Azure resources and cannot be included in a storage account. The following illustration shows a typical arrangement: Blobs, Files, Queues, and Tables are inside storage accounts, while other services are not.



https://www.c-sharpcorner.com/article/what-is-microsoft-azure-storage/

Question 68: Skipped

Scenario: You are working in an Azure Databricks workspace and you want to filter based on the end of a column value using the Column Class. Specifically, you are looking at a column named verb and filtered by words ending with *"ing"*.

Which command filters based on the end of a column value as required?

- ○ `df.filter(col("verb").endswith("ing"))` (Correct)
- ○ `df.filter().col("verb").like("%ing")`
- ○ `df.filter("verb like ‘_ing'")`
- ○ `df.filter("verb like '%ing'")`

Explanation

The Column Class supports both the `endswith()` method and the like() method (example - `col("verb").like("%ing")`).

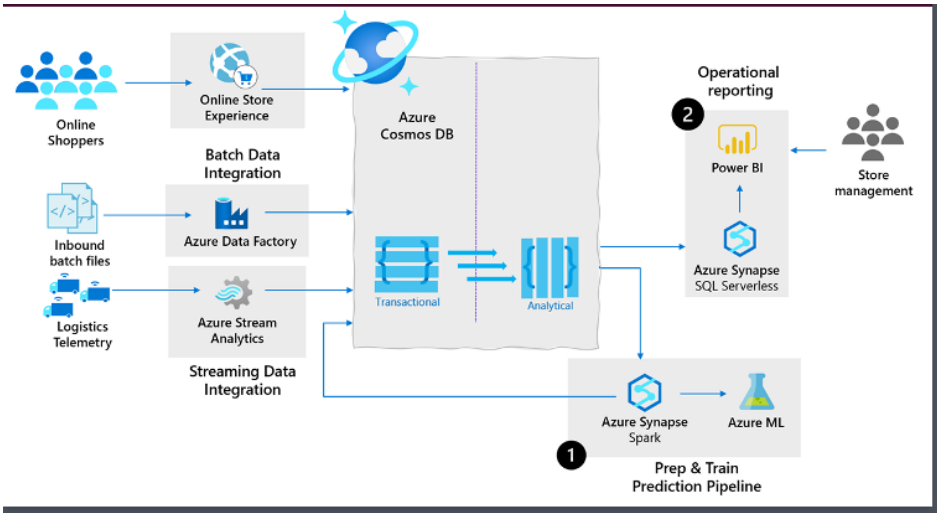
https://docs.databricks.com/spark/latest/dataframes-datasets/introduction-to-dataframes-python.html

Question 69: **Skipped**

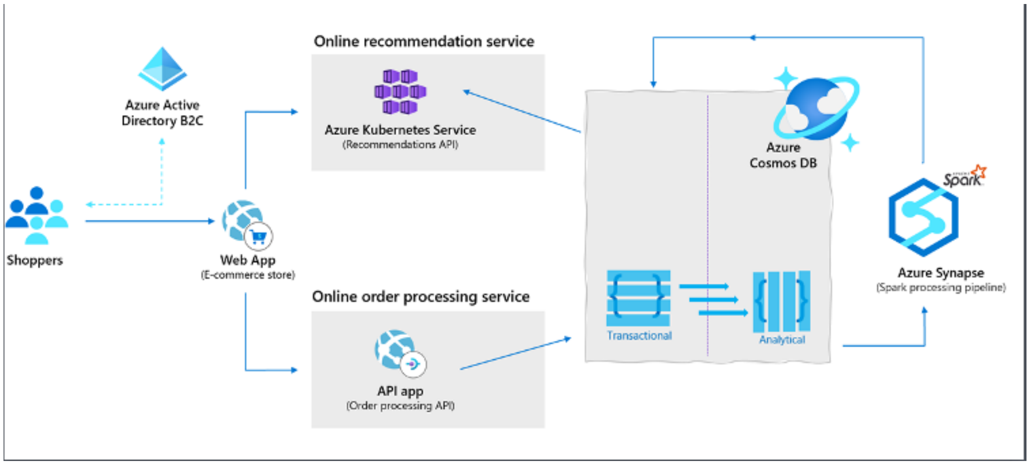
Scenario: You are working at Jungle.com which is a web-based retailer which needs to perform real-time basket analysis to make product recommendations to customers who are about to purchase products. This increased revenues for the organization as the provided targeted suggestions at the point of sale.

Review the following architecture designs.

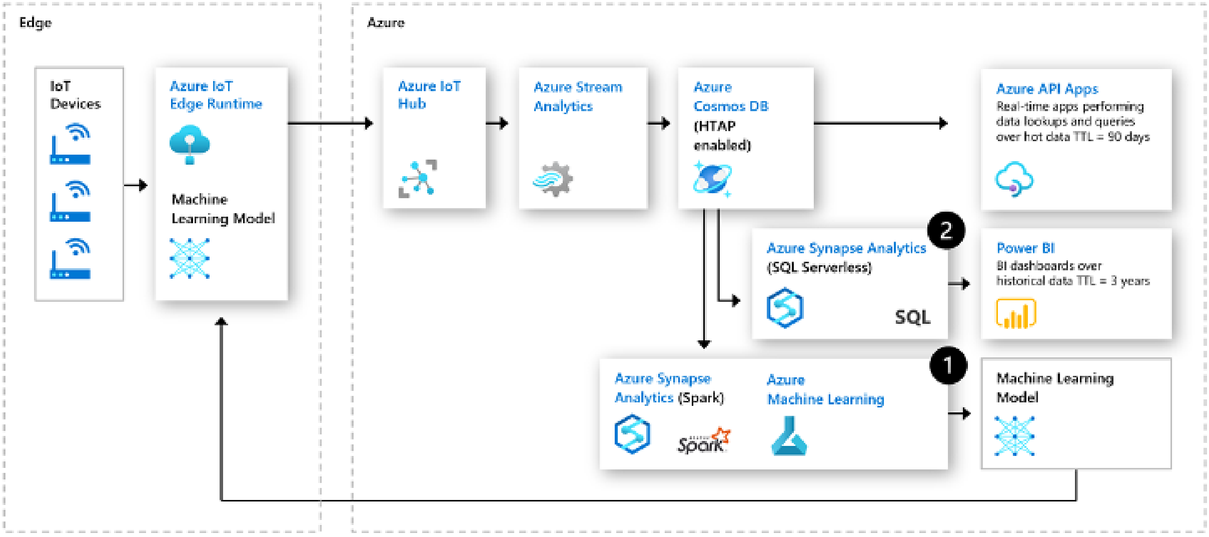
Design A:



Design B:



Design C:


Edge
IoT Devices
Azure IoT Edge Runtime
Machine Learning Model
Azure
Azure IoT Hub
Azure Stream Analytics
Azure Cosmos DB (HTAP enabled)
Azure API Apps
Real-time apps performing data lookups and queries over hot data TTL = 90 days
2
Azure Synapse Analytics (SQL Serverless)
SQL
Power BI
BI dashboards over historical data TTL = 3 years
1
Azure Synapse Analytics (Spark)
Spark
Azure Machine Learning
Machine Learning Model

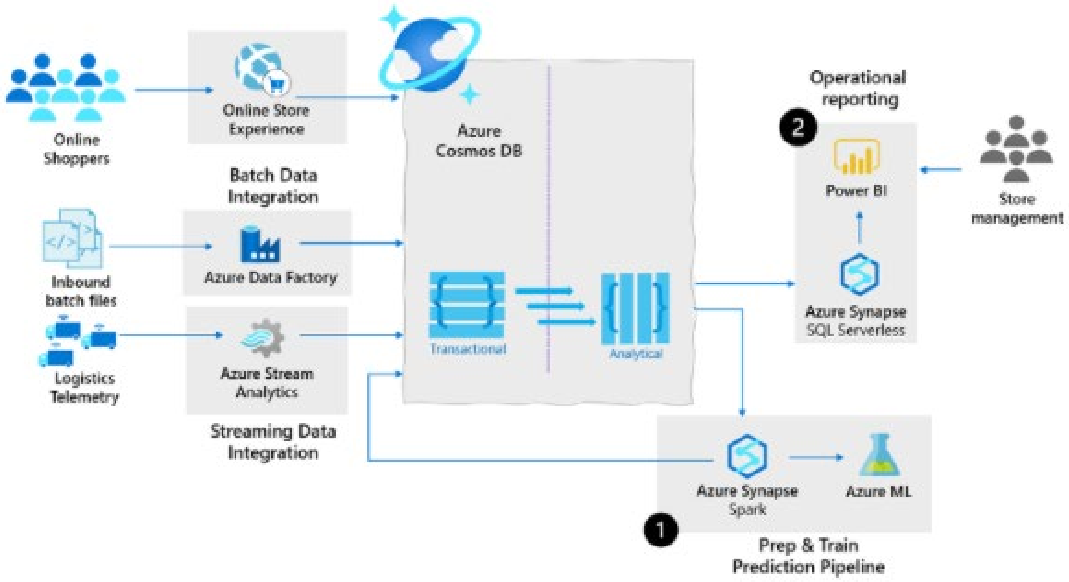
Which design would be best suited for the need?

- ○ Design C
- ○ None of the listed options
- ○ Design B (Correct)
- ○ Design A

Explanation
Supply chain analytics, forecasting and reporting.

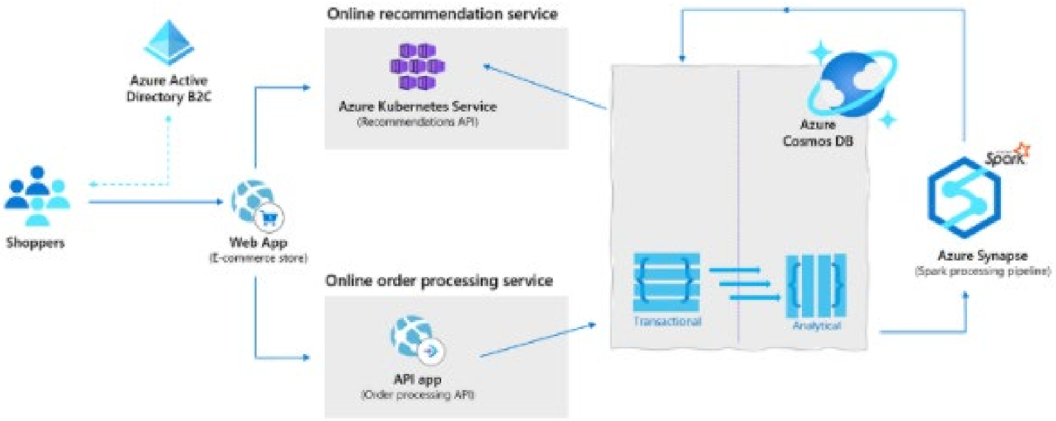
With supply chains generating increasing volumes of operational data every minute for orders, shipments and sales transactions, manufactures and retailers need an operational database that can scale to handle the data volumes as well as an analytical platform to get to a level of real-time contextual intelligence to stay ahead of the curve.

Azure Synapse Link for Cosmos DB allows these organizations to store data from their sales systems, ingest real-time telemetry data from in vehicle systems and integrate date from their ERP systems into a common operational store in Azure Cosmos DB and then leverage the data from Synapse analytics to enable both predictive analytics scenarios such as stock out monitoring and supply chain bottleneck management (1) in addition to enabling operational reporting directly on their operation data using standard reporting tools such as Power BI (2).



Retail real-time personalization.

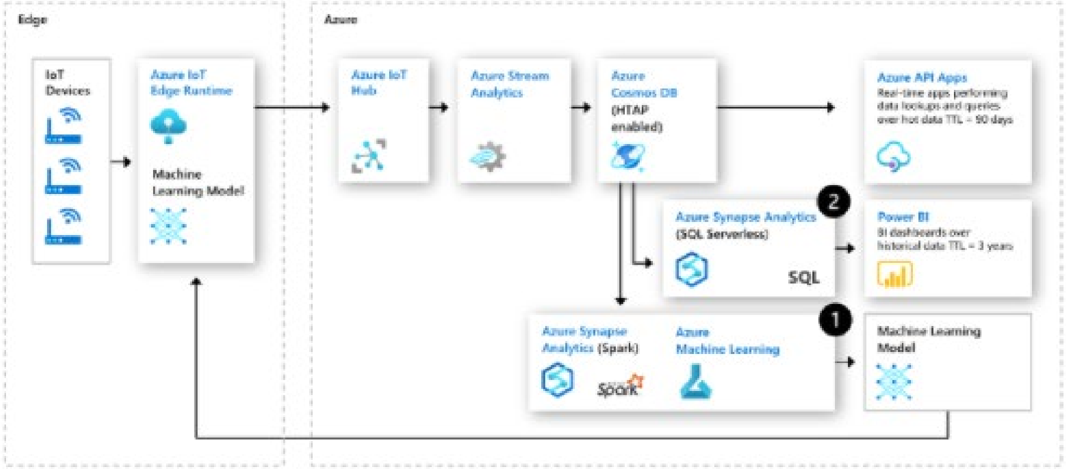
In retail, many web-based retailers will perform real-time basket analysis to make product recommendations to customers who are about to purchase products. This increased revenues for these organizations as the provided targeted suggestions at the point of sales.



Predictive maintenance using anomaly detection with IOT

Industrial IOT innovations have drastically reduced downtimes of machinery and increased overall efficiency across all fields of industry. One of such innovations is predictive maintenance analytics for machinery at the edge of the cloud.

The following architecture leverages the cloud native HTAP capabilities of Azure Synapse Link for Azure Cosmos DB in IoT predictive maintenance:


Edge
IoT Devices
Azure IoT Edge Runtime
Machine Learning Model
Azure
Azure IoT Hub
Azure Stream Analytics
Azure Cosmos DB (HTAP enabled)
Azure API Apps
Real-time apps performing data lookups and queries over hot data TTL = 90 days
2
Azure Synapse Analytics (SQL Serverless)
SQL
Power BI
BI dashboards over historical data TTL = 3 years
1
Azure Synapse Analytics (Spark)
Spark
Azure Machine Learning
Machine Learning Model

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

Question 70: **Skipped**

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

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

[?] leverages the capabilities of Azure Data Factory and is the cloud-based ETL and data integration service that allows you to create data-driven workflows for orchestrating data movement and transforming data at scale. Using [?], you can create and schedule data-driven workflows that can ingest data from disparate data stores.

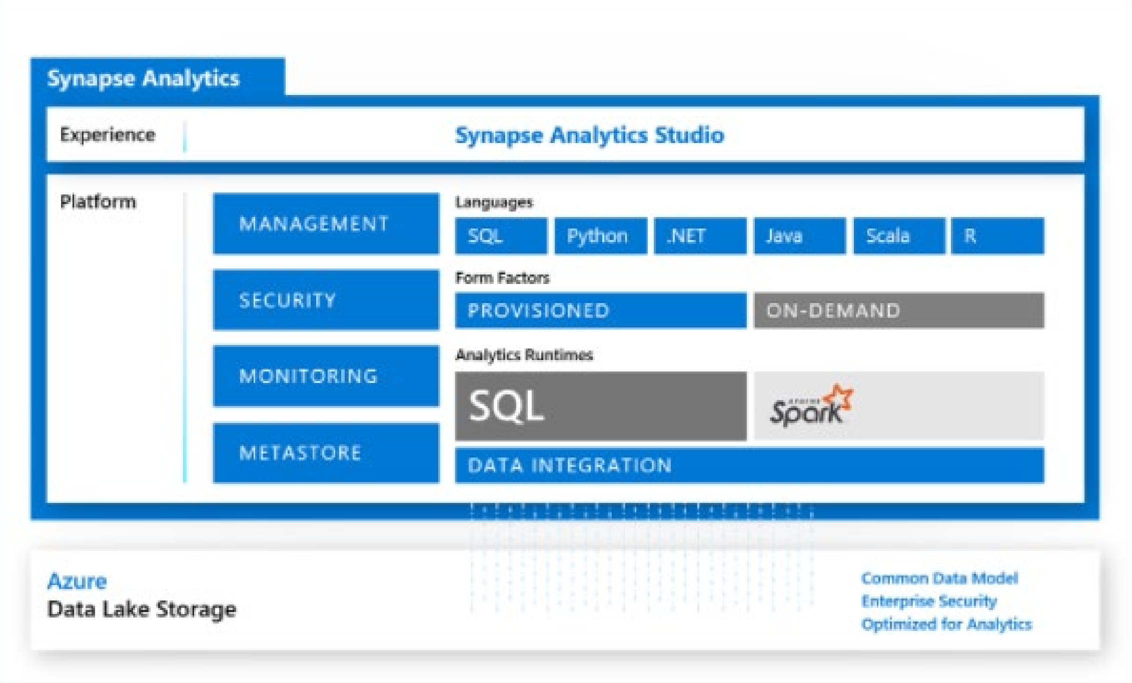
- ○ Apache Spark for Azure Synapse
- ○ Azure Synapse Link
- ○ Azure Synapse Pipelines (Correct)
- ○ Azure Synapse SQL
- ○ Azure Cosmos DB

Explanation

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

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

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



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

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

Data integration to integrate your data with Azure Synapse Pipelines

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

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

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

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

Question 71: **Skipped**

Scenario: You are working at a bank setting up a database which will be used by all employee-levels of the bank. At the moment, you are setting up permissions for service representatives in a call centre.

Often, due to compliance, the caller has to identify themselves by giving them the last four digits of their credit card number that they may have an issue with. These data items cannot be fully exposed to the service representative in that call centre.

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

- ○ Dynamic Data Masking (Correct)
- ○ Column-level security
- ○ Table-level security
- ○ Row-level security

Explanation

If you would define a masking rule, that masks all but the last four digits for example of that credit card number, you would get a query that only gives as a result the last four digits of the credit card number.

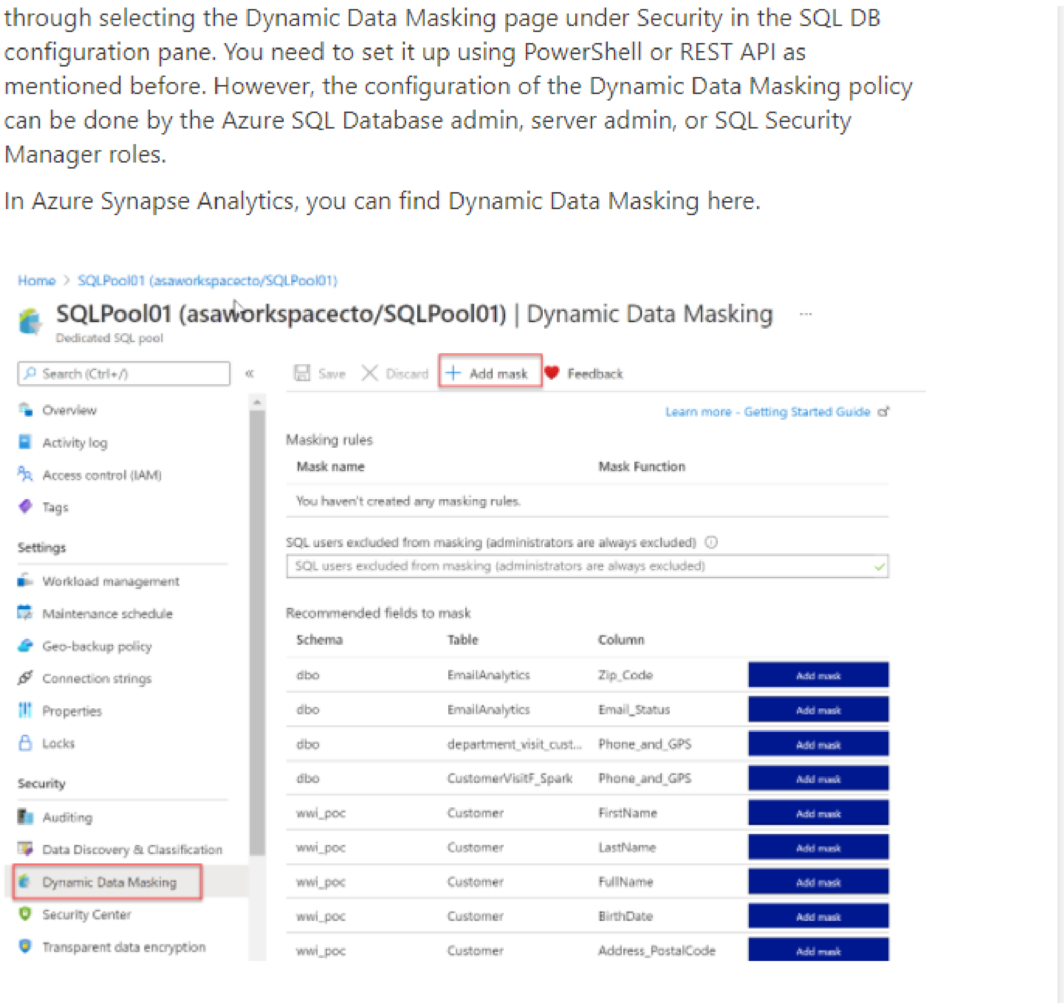
Dynamic Data Masking

Azure SQL Database, Azure SQL Managed Instance, and Azure Synapse Analytics support Dynamic Data Masking. It's all in the name, Dynamic Data Masking is masking and ensures limited data exposure to non-privileged users, such that they can't see it. It also helps you in preventing unauthorized access to sensitive data. The way Dynamic Data Masking does it, is helping customers to designate how much of the sensitive data to reveal such that it has minimal impact on the application layer. Dynamic Data Masking is a policy-based security feature. It will hide the sensitive data in a result set of a query that runs over designated database fields. However, the data in the database will not be changed.

Let's give you an example how it works. Let's say you work at a bank as a service representative in a call centre. Sometime, due to compliance, the caller has to identify themselves by giving them several digits of their credit card number that they might have an issue with. However, these data items, should not be fully exposed to the service representative in that call centre, answering the call. If you would define a masking rule, that masks all but the last four digits for example of that credit card number, you would get a query that only gives as a result the last four digits of the credit card number. If the caller, for example, also had to provide the representative with personal information, that should not be seen by the developer that can query the production environments in order to troubleshoot, you should appropriately mask data in order to protect the given personal data such that compliance is not violated.

For Azure Synapse Analytics, the way to set up a Dynamic Data Masking policy is using PowerShell or the REST API. Bear in mind that it won't be possible for Azure Synapse Analytics to set the Dynamic Data Masking policy in the Azure portal through selecting the Dynamic Data Masking page under Security in the SQL DB configuration pane. You need to set it up using PowerShell or REST API as mentioned before. However, the configuration of the Dynamic Data Masking policy can be done by the Azure SQL Database admin, server admin, or SQL Security Manager roles.

In Azure Synapse Analytics, you can find Dynamic Data Masking here.



through selecting the Dynamic Data Masking page under Security in the SQL DB configuration pane. You need to set it up using PowerShell or REST API as mentioned before. However, the configuration of the Dynamic Data Masking policy can be done by the Azure SQL Database admin, server admin, or SQL Security Manager roles.

In Azure Synapse Analytics, you can find Dynamic Data Masking here.

Masking rules

Mask name	Mask Function
You haven't created any masking rules.	

SQL users excluded from masking (administrators are always excluded)

SQL users excluded from masking (administrators are always excluded)

Recommended fields to mask

Schema	Table	Column	
dbo	EmailAnalytics	Zip_Code	Add mask
dbo	EmailAnalytics	Email_Status	Add mask
dbo	department_visit_cust...	Phone_and_GPS	Add mask
dbo	CustomerVisitF_Spark	Phone_and_GPS	Add mask
wwi_poc	Customer	FirstName	Add mask
wwi_poc	Customer	LastName	Add mask
wwi_poc	Customer	FullName	Add mask
wwi_poc	Customer	BirthDate	Add mask
wwi_poc	Customer	Address_PostalCode	Add mask

Looking into Dynamic Data Masking Policies:

- **SQL users are excluded from masking**

A couple of SQL users or Azure AD identities can get unmasked data in the SQL query results. Users with administrator privileges are always excluded from masking, and see the original data without any mask.

- **Masking rules** - Masking rules are a set of rules that define the designated fields to be masked including the masking function that is used. The designated fields can be defined using a database schema name, table name, and column name.

- **Masking functions** - Masking functions are a set of methods that control the exposure of data for different scenarios.

Dynamic Data Masking for your database in Azure Synapse Analytics using PowerShell cmdlets

- Data masking policies

- `Get-AzSqlDatabaseDataMaskingPolicy`

The `Get-AzSqlDatabaseDataMaskingPolicy` gets the data masking policy for a database.

The syntax for the `Get-AzSqlDatabaseDataMaskingPolicy` in PowerShell is as follows:

```
PowerShell

Get-AzSqlDatabaseDataMaskingPolicy [-ServerName] <String> [-DatabaseName] <String
>

[-ResourceGroupName] <String> [-DefaultProfile <IAzureContextContainer>] [-WhatIf
] [-Confirm]

[<CommonParameters>]
```

What the `Get-AzSqlDatabaseDataMaskingPolicy` cmdlet does, is getting the data masking policy of an Azure SQL database.

To use this cmdlet in PowerShell, you'd have to specify the following parameters to identify the database:

- *ResourceGroupName*: name of the resource group you deployed the database in

- *ServerName*: sql server name

- *DatabaseName* : name of the database

This cmdlet is also supported by the SQL Server Stretch Database service on Azure.

- `Set-AzSqlDatabaseDataMaskingPolicy`

The `Set-AzSqlDatabaseDataMaskingPolicy` sets data masking for a database.

The syntax for the `Set-AzSqlDatabaseDataMaskingPolicy` in PowerShell is as follows:

```
PowerShell

Set-AzSqlDatabaseDataMaskingPolicy [-PassThru] [-PrivilegedUsers <String>] [-DataMaskingState <String>]

[-ServerName] <String> [-DatabaseName] <String> [-ResourceGroupName] <String>

[-DefaultProfile <IAzureContextContainer>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

What the `Set-AzSqlDatabaseDataMaskingPolicy` cmdlet does is setting the data masking policy for an Azure SQL database.

To use this cmdlet in PowerShell, you'd have to specify the following parameters to identify the database:

- *ResourceGroupName*: name of the resource group that you deployed the database in

- *ServerName* : sql server name

- *DatabaseName* : name of the database

In addition, you will need to set the *DataMaskingState* parameter to specify whether data masking operations are enabled or disabled.

If the cmdlet succeeds and the *PassThru* parameter is used, it will return an object describing the current data masking policy in addition to the database identifiers.

Database identifiers can include, **ResourceGroupName**, **ServerName**, and **DatabaseName**.

This cmdlet is also supported by the SQL Server Stretch Database service on Azure.

- Data masking rules

- `Get-AzSqlDatabaseDataMaskingRule`

The `Get-AzSqlDatabaseDataMaskingRule` Gets the data masking rules from a database.

The syntax for the `Get-AzSqlDatabaseDataMaskingRule` in PowerShell is as follows:

```
PowerShell

Get-AzSqlDatabaseDataMaskingRule [-SchemaName <String>] [-TableName <String>] [-C
olumnName <String>]

[-ServerName] <String> [-DatabaseName] <String> [-ResourceGroupName] <String>

[-DefaultProfile <IAzureContextContainer>] [-WhatIf] [-Confirm] [<CommonParameter
s>]
```

What the `Get-AzSqlDatabaseDataMaskingRule` cmdlet does it getting either a specific data masking rule or all of the data masking rules for an Azure SQL database.

To use the cmdlet in PowerShell, you'd have to specify the following parameters to identify the database:

To use this cmdlet in PowerShell, you'd have to specify the following parameters to identify the database:

- *ResourceGroupName*: name of the resource group that you deployed the database in

- *ServerName* : sql server name

- *DatabaseName* : name of the database

You'd also have to specify the *RuleId* parameter to specify which rule this cmdlet returns.

If you do not provide *RuleId*, all the data masking rules for that Azure SQL database are returned.

This cmdlet is also supported by the SQL Server Stretch Database service on Azure.

- `New-AzSqlDatabaseDataMaskingRule`

The `New-AzSqlDatabaseDataMaskingRule` creates a data masking rule for a database.

The syntax for the `New-AzSqlDatabaseDataMaskingRule` in PowerShell is as follows:

```
PowerShell
```

```
New-AzSqlDatabaseDataMaskingRule -MaskingFunction <String> [-PrefixSize <UInt32>]
[-ReplacementString <String>]

[-SuffixSize <UInt32>] [-NumberFrom <Double>] [-NumberTo <Double>] [-PassThru] -S
chemaName <String>

-TableName <String> -ColumnName <String> [-ServerName] <String> [-DatabaseName] <
String>

[-ResourceGroupName] <String> [-DefaultProfile <IAzureContextContainer>] [-WhatIf
] [-Confirm]

[<CommonParameters>]
```

What the `New-AzSqlDatabaseDataMaskingRule` cmdlet does is creating a data masking rule for an Azure SQL database.

To use this cmdlet in PowerShell, you'd have to specify the following parameters to identify the rule:

• *ResourceGroupName*: name of the resource group that you deployed the database in

• *ServerName* : sql server name

• *DatabaseName* : name of the database

Providing the *TableName* and *ColumnName* is necessary in order to specify the target of the rule.

The *MaskingFunction* parameter is necessary to define how the data is masked.

If *MaskingFunction* has a value of Number or Text, you can specify the *NumberFrom* and *NumberTo* parameters, for number masking, or the *PrefixSize*, *ReplacementString*, and *SuffixSize* for text masking.

If the command succeeds and the *PassThru* parameter is used, the cmdlet returns an object describing the data masking rule properties in addition to the rule identifiers.

Rule identifiers can be, for example, *ResourceGroupName*, *ServerName*, *DatabaseName*, and *RuleID*.

This cmdlet is also supported by the SQL Server Stretch Database service on Azure.

• `Remove-AzSqlDatabaseDataMaskingRule`

The `Remove-AzSqlDatabaseDataMaskingRule` removes a data masking rule from a database.

The syntax for the `Remove-AzSqlDatabaseDataMaskingRule` in PowerShell is as follows:

```
PowerShell

Remove-AzSqlDatabaseDataMaskingRule [-PassThru] [-Force] -SchemaName <String> -TableName <String>

-ColumnName <String> [-ServerName] <String> [-DatabaseName] <String> [-ResourceGroupName] <String>

[-DefaultProfile <IAzureContextContainer>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

What the `Remove-AzSqlDatabaseDataMaskingRule` cmdlet does, is it removes a specific data masking rule from an Azure SQL database.

To use this cmdlet in PowerShell, you'd have to specify the following parameters to identify the rule that needs to be removed:

• *ResourceGroupName*: name of the resource group that you deployed the database in

• *ServerName* : sql server name

• *DatabaseName* : name of the database

• *RuleId* : identifier of the rule

This cmdlet is also supported by the SQL Server Stretch Database service on Azure.

• `Set-AzSqlDatabaseDataMaskingRule`

The `Set-AzSqlDatabaseDataMaskingRule` Sets the properties of a data masking rule for a database.

The syntax for the `Set-AzSqlDatabaseDataMaskingRule` in PowerShell is as follows:

```
PowerShell

Set-AzSqlDatabaseDataMaskingRule [-MaskingFunction <String>] [-PrefixSize <UInt32>]

[-ReplacementString <String>] [-SuffixSize <UInt32>] [-NumberFrom <Double>] [-NumberTo <Double>] [-PassThru]

-SchemaName <String> -TableName <String> -ColumnName <String> [-ServerName] <String> [-DatabaseName] <String>

[-ResourceGroupName] <String> [-DefaultProfile <IAzureContextContainer>] [-WhatIf] [-Confirm]
```

```
[<CommonParameters>]
```

What the `Set-AzSqlDatabaseDataMaskingRule` cmdlet does is setting a data masking rule for an Azure SQL database.

To use this cmdlet in PowerShell, you'd have to specify the following parameters to identify the rule:

- *ResourceGroupName*: name of the resource group that you deployed the database in

- *ServerName* : sql server name

- *DatabaseName* : name of the database

- *RuleId* : identifier of the rule

You can provide any of the parameters of *SchemaName*, *TableName*, and *ColumnName* to retarget the rule.

Specify the *MaskingFunction* parameter to modify how the data is masked.

If you specify a value of Number or Text for *MaskingFunction*, you can specify the *NumberFrom* and *NumberTo* parameters for number masking or the *PrefixSize*, *ReplacementString*, and *SuffixSize* parameters for text masking.

If the command succeeds, and if you specify the *PassThru* parameter, the cmdlet returns an object that describes the data masking rule properties and the rule identifiers.

Rule identifiers can be, **ResourceGroupName**, **ServerName**, **DatabaseName**, and **RuleId**.

This cmdlet is also supported by the SQL Server Stretch Database service on Azure.

Set up Dynamic Data Masking for your database in Azure Synapse Analytics using the REST API

For setting up Dynamic Data Masking in Azure Synapse Analytics, the other possibility is make use of the REST API.

It will enable to programmatically manage data masking policy and rules.

The REST API will support the following operations:

- Data masking policies

- Create Or Update

The Create Or Update masking policy using the REST API will create or update a database data masking policy.

In HTTP the following request can be made:

```
HTTP

PUT https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Sql/servers/{serverName}/databases/{databaseName}/dataMaskingPolicies/Default?api-version=2014-04-01
```

The following parameters need to be passed through:

- *SubscriptionID*: the ID of the subscription

- *ResourceGroupName*: name of the resource group that you deployed the database in

- *ServerName* : sql server name

- *DatabaseName* : name of the database

- *dataMaskingPolicyName*: the name of the data masking policy

- *api version*: version of the api that is used.

- Get

The Get policy, Gets a database data masking policy.

In HTTP the following request can be made:

```
HTTP

GET https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Sql/servers/{serverName}/databases/{databaseName}/dataMaskingPolicies/Default?api-version=2014-04-01
```

The following parameters need to be passed through:

- *SubscriptionID*: the ID of the subscription

- *ResourceGroupName*: name of the resource group that you deployed the database in

- *ServerName* : sql server name

- *DatabaseName* : name of the database

- *dataMaskingPolicyName*: the name of the data masking policy

- *api version*: version of the api that is used.

- Data masking rules

- Create Or Update

The Create or Update masking rule creates or updates a database data masking rule.

In HTTP the following request can be made:

```
HTTP

PUT https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Sql/servers/{serverName}/databases/{databaseName}/dataMaskingPolicies/Default/rules/{dataMaskingRuleName}?api-version=2014-04-01
```

The following parameters need to be passed through:

- *SubscriptionID*: the ID of the subscription

- *ResourceGroupName*: name of the resource group that you deployed the database in

- *ServerName* : sql server name

- *DatabaseName* : name of the database

- *dataMaskingPolicyName*: the name of the data masking policy

- *dataMaskingRuleName*: the name of the rule for data masking

- *api version*: version of the api that is used.

- List By Database

The List By Database request gets a list of database data masking rules.

In HTTP the following request can be made:

```
HTTP

GET https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Sql/servers/{serverName}/databases/{databaseName}/dataMaskingPolicies/Default/rules?api-version=2014-04-01
```

The following parameters need to be passed through:

- *SubscriptionID*: the ID of the subscription

- *ResourceGroupName*: name of the resource group that you deployed the database in

- *ServerName* : sql server name

- *DatabaseName* : name of the database

- *dataMaskingPolicyName*: the name of the data masking policy

- *api version*: version of the api that is used.

https://docs.microsoft.com/en-us/azure/azure-sql/database/dynamic-data-masking-overview

Question 72: **Skipped**

Which transformation in the Mapping Data Flow is used to routes data rows to different streams based on matching conditions?

- ○ Alter row
- ○ Select
- ○ Conditional Split **(Correct)**
- ○ Lookup
- ○ Multiple inputs/outputs
- ○ Derived column

Explanation

A Conditional Split transformation routes data rows to different streams based on matching conditions. The conditional split transformation is similar to a CASE decision structure in a programming language.

https://docs.microsoft.com/en-us/azure/data-factory/data-flow-conditional-split

Transforming data using Mapping Data Flow

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

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

• Schema modifier transformations

• Row modifier transformations

• Multiple inputs/outputs transformations

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

Name & Category: Aggregate - Schema modifier

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

Name & Category: Alter row - Row modifier

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

Name & Category: Conditional split - Multiple inputs/outputs

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

Name & Category: Derived column - Schema modifier

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

Name & Category: Exists - Multiple inputs/outputs

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

Name & Category: Filter - Row modifier

Description: Filter a row based upon a condition.

Name & Category: Flatten - Schema modifier

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

Name & Category: Join - Multiple inputs/outputs

Description: Combine data from two sources or streams.

Name & Category: Lookup - Multiple inputs/outputs

Description: Enables you to reference data from another source.

Name & Category: New branch - Multiple inputs/outputs

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

Name & Category: Pivot - Schema modifier

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

Name & Category: Select - Schema modifier

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

Name & Category: Sink – N/A

Description: A final destination for your data.

Name & Category: Sort - Row modifier

Description: Sort incoming rows on the current data stream.

Name & Category: Source – N/A

Description: A data source for the data flow.

Name & Category: Surrogate key - Schema modifier

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

Name & Category: Union - Multiple inputs/outputs

Description: Combine multiple data streams vertically.

Name & Category: Unpivot - Schema modifier

Description: Pivot columns into row values.

Name & Category: Window - Schema modifier

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

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

Question 73: **Skipped**

Dynamic Management Views provide a programmatic experience for monitoring the Azure Synapse Analytics SQL pool activity by using the Transact-SQL language. What type of information or assistance do the views provide? (Select all that apply)

- □ SQL execution requests and queries **(Correct)**
- □ Troubleshoot workload performance bottlenecks **(Correct)**
- □ Connection information and activity **(Correct)**
- □ Encryption deficiencies
- □ Identify workload performance bottlenecks **(Correct)**
- □ Data movement service activity **(Correct)**
- □ Resource blocking and locking activity **(Correct)**

Explanation

Dynamic Management Views provide a programmatic experience for monitoring the Azure Synapse Analytics SQL pool activity by using the Transact-SQL language. The views that are provided, not only enable you to troubleshoot and identify performance bottlenecks with the workloads working on your system, but they are also used by other services such as Azure Advisor to provide recommendations about Azure Synapse Analytics.

There are over 90 Dynamic Management Views that can queried against dedicated SQL pools to retrieve information about the following areas of the service:

- Connection information and activity

- SQL execution requests and queries

- Index and statistics information

- Resource blocking and locking activity
- Data movement service activity
- Errors

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

Question 74: **Skipped**

You can integrate your Azure Synapse Analytics workspace with a new Power BI workspace so that you can get you data from within Azure Synapse Analytics visualized in a Power BI report or dashboard.

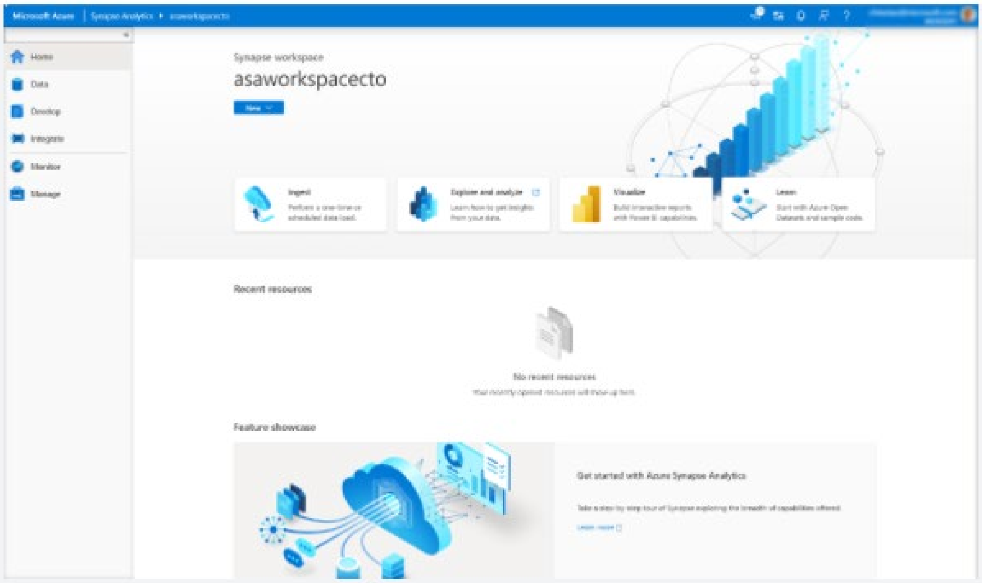
Which icon should you click on the home page of Azure Synapse Studio to begin the integration?

- ○ Ingest
- ○ Explore and analyze
- ○ None of the listed options (Correct)
- ○ Import
- ○ Connect BI

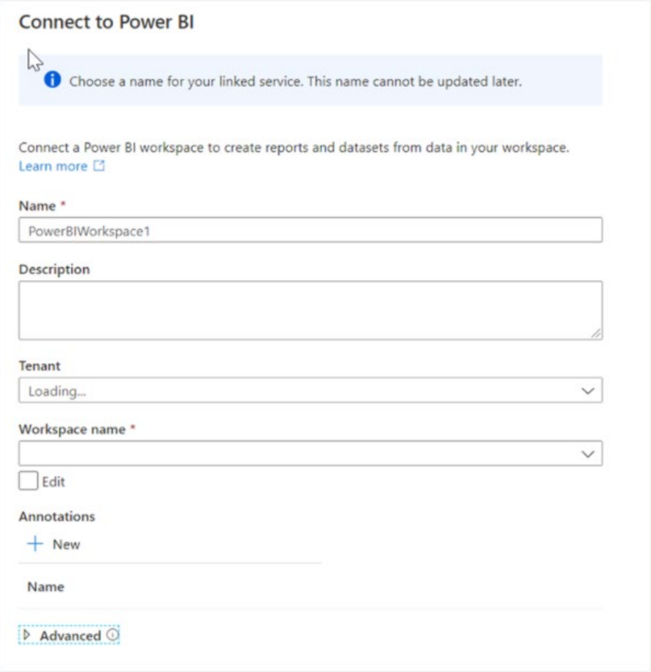
Explanation

You can integrate your Azure Synapse Analytics workspace with a new Power BI workspace so that you can get you data from within Azure Synapse Analytics visualized in a Power BI report or dashboard.

You can perform this step by clicking on the visualize icon on the home page of Azure Synapse Studio.


Microsoft Azure | Synapse Analytics ▸ asaworkspacecto
Home
Data
Develop
Integrate
Monitor
Manage
Synapse workspace
asaworkspacecto
New
Ingest
Perform a one-time or scheduled data load.
Explore and analyze
Learn how to get insights from your data.
Visualize
Build interactive reports with Power BI capabilities.
Learn
Start with Azure Open Datasets and sample code.
Recent resources
No recent resources
Your recently opened resources will show up here.
Feature showcase
Get started with Azure Synapse Analytics
Take a step-by-step tour of Synapse exploring the breadth of capabilities offered.
Learn more

Which will bring up the Connect to Power BI screen.



Connect to Power BI

Choose a name for your linked service. This name cannot be updated later.

Connect a Power BI workspace to create reports and datasets from data in your workspace.
Learn more

Name *
PowerBIWorkspace1

Description

Tenant
Loading...

Workspace name *

□ Edit

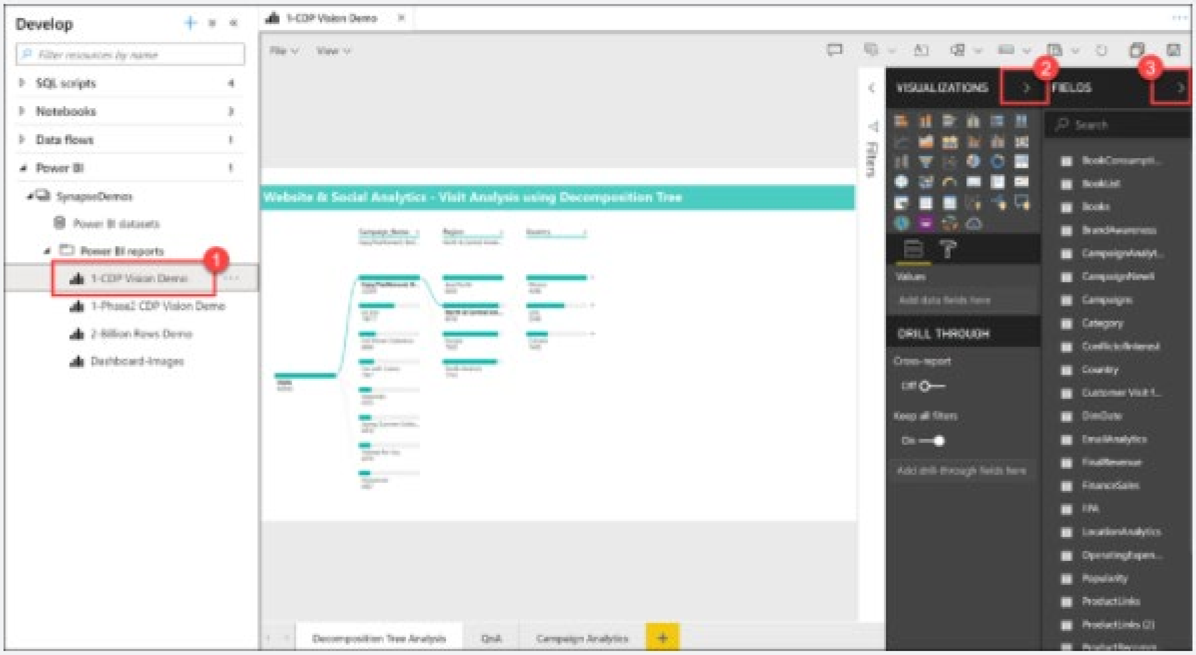
Annotations
+ New

Name

Advanced

From here you can define a name and description for the Power BI Workspace. Then you would select the Tenant and Workspace name. Once you have connected to your workspace, you will be able to access the existing reports in the Power BI workspace in the Develop hub in Azure Synapse Studio.

Expand Power BI, expand SynapseDemos, expand Power BI reports, then select **1-CDP Vision Demo (1)**. Select the arrows to collapse the **Visualizations pane **(2)** and the **Fields** pane **(3)** to increase the report size.



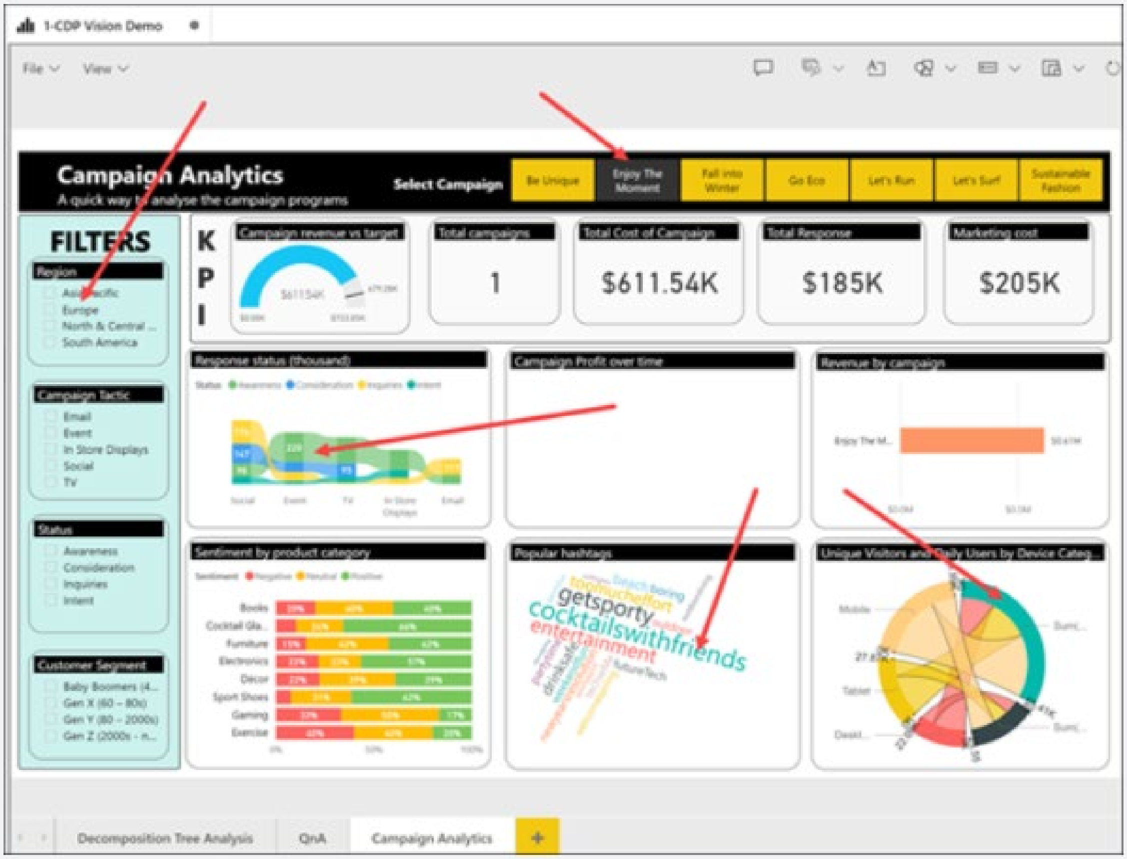
As you can see, we can create, edit, and view Power BI reports from within Synapse Studio! As a business analyst, data engineer, or developer, you no longer need to open another browser window, sign in to Power BI, and toggle back and forth between environments.

Select a **Campaign Name** and **Region** within the **Decomposition Tree Analysis** tab to explore the data. If you hover over an item, you will see a tool tip.

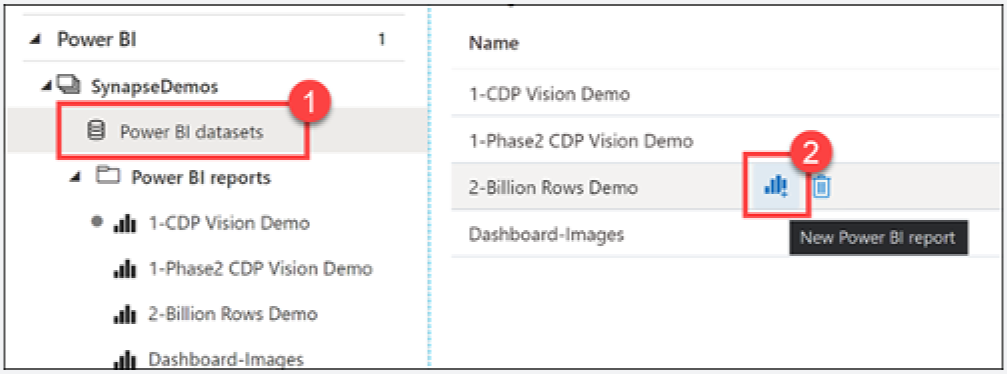
Select the **Campaign Analytics** tab at the bottom of the report.

The Campaign Analytics report combines data from the various data sources to create a compelling visualization of valuable data within an interactive interface.

You can select various filters, campaigns, and chart values to filter the results. Select an item to for the second time to deselect it.

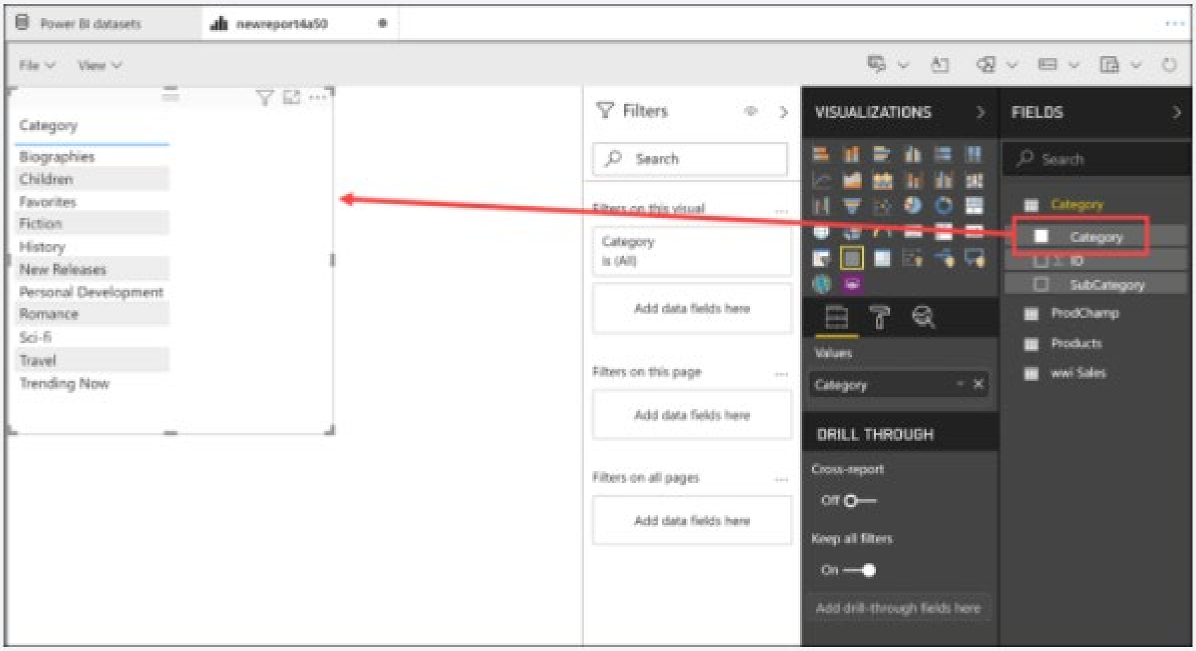


Select **Power BI datasets (1)** in the left-hand menu, hover over the **2-Billion Rows Demo** dataset and select the **New Power BI report** icon **(2)**.

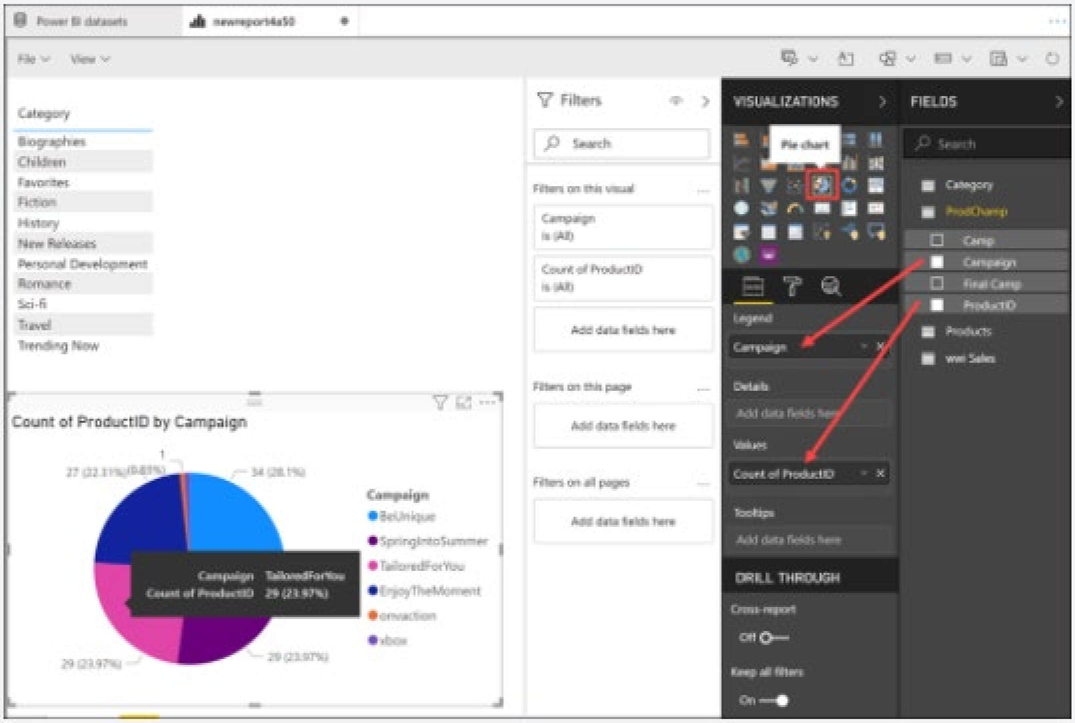


Here is how we can create a brand new Power BI report from a dataset that is part of the linked Power BI workspace, from within Synapse Studio.

Expand the Category table, then **drag-and-drop** the **Category** field on to the report canvas. This creates a new Table visualization that shows the categories.



Select a blank area on the report canvas to deselect the table, then select the **Pie chart** visualization.



Expand the ProdChamp table. Drag **Campaign** onto the **Legend** field, then drag **ProductID** onto the **Values** field. Resize the pie chart and hover over the pie slices to see the tool tips.

We have very quickly created a new Power BI report, using data stored within our Synapse Analytics workspace, without ever leaving the studio.

https://docs.microsoft.com/en-us/azure/synapse-analytics/quickstart-power-bi

Question 75: **Skipped**

What does the `APPROX_COUNT_DISTINCT` Transact-SQL function do?

- ○

  Approximate execution using Hyperlog accuracy

  (Correct)

- ○

  Approximate count on distinct executions within a specified time period on a specific endpoint.

- ○

  Calculates the approximate number of distinct records in a non-relational database.

- ○

  None of the listed options.

- ○

  Calculates the approximate number of distinct records in a relational database.

Explanation

It is not uncommon for data engineers, data analysts, and data scientists alike to perform exploratory data analysis to gain an understanding of the data that they are working with. Exploratory data analysis can involve querying metadata about the data that is stored within the database, to running queries to provide a statistics information about the data such as average values for a column, through to distinct counts. Some of the activities can be time consuming, especially on large data sets.

For example, performing a distinct count of values in a Billion plus row table can be an expensive operation that takes time to resolve. As exploratory data analysis sometime doesn't require accurate information, there is a solution.

Azure Synapse Analytics supports Approximate execution using Hyperlog accuracy to reduce latency when executing queries with large datasets. Approximate execution is used to speed up the execution of queries with a compromise for a small reduction in accuracy. So if it takes too long to get basic information about the data in a large data set as you are exploring data of a big data set, then you can use the `HyperLogLog` accuracy and will return a result with a 2% accuracy of true cardinality on average. This is done by using the `APPROX_COUNT_DISTINCT` Transact-SQL function.

https://www.slideshare.net/jamserra/azure-synapse-analytics-overview

Question 76: **Skipped**

There are a range of network security steps that you should consider to secure Azure Synapse Analytics. One of the first aspects that you will consider is securing access to the service itself. This can be achieved by creating the following network objects including:

• Firewall rules

• Virtual networks

• Private endpoints

Which of the following are benefits of using a managed workspace virtual network? (Select all that apply)

- □ You don't have to configure inbound NSG rules on your own Virtual Networks to allow Azure Synapse management traffic to enter your Virtual Network. **(Correct)**
- □ You don't need to create a subnet for your Spark clusters based on peak load. **(Correct)**
- □ It ensures that your workspace is a consolidated network with your other workspaces.
- □ With a Managed workspace Virtual Network, you can offload the burden of managing the Virtual Network to Azure Synapse. **(Correct)**
- □ Managed workspace Virtual Network along with Managed private endpoints protects against data exfiltration. **(Correct)**

Explanation

There are a range of network security steps that you should consider to secure Azure Synapse Analytics. One of the first aspects that you will consider is securing access to the service itself. This can be achieved by creating the following network objects including:

• Firewall rules

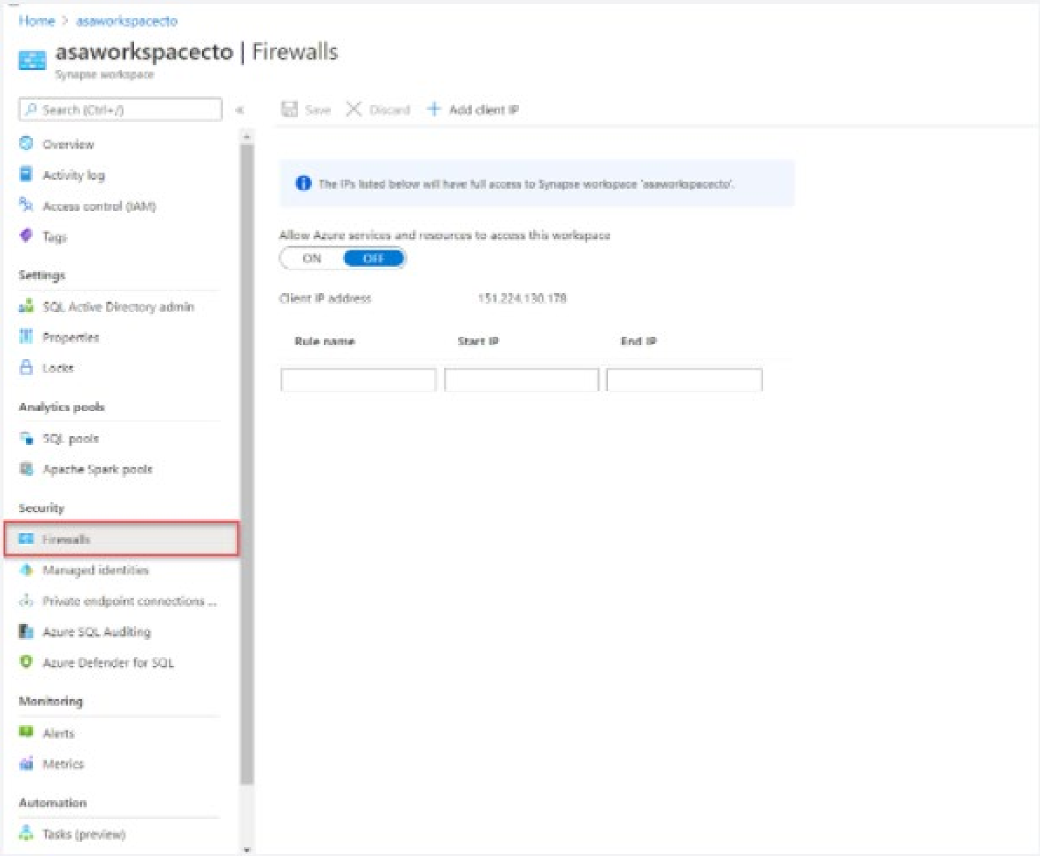
• Virtual networks

- Private endpoints

Firewall rules

Firewall rules enable you to define the type of traffic that is allowed or denied access to an Azure Synapse workspace using the originating IP address of the client that is trying to access the Azure Synapse Workspace. IP firewall rules configured at the workspace level apply to all public endpoints of the workspace including dedicated SQL pools, serverless SQL pool, and the development endpoint.

You can choose to allow connections from all IP addresses as you are creating the Azure Synapse Workspaces, although this is not recommended as it does not allow for control access to the workspace. Instead, within the Azure portal, you can configure specific IP address ranges and associate them with a rule name so that you have greater control.



The IPs listed below will have full access to Synapse workspace 'asaworkspacecto'.

Allow Azure services and resources to access this workspace
○ ON ● OFF

Client IP address 151.224.130.178

Rule name	Start IP	End IP

Make sure that the firewall on your network and local computer allows outgoing communication on TCP ports 80, 443 and 1443 for Synapse Studio.

Also, you need to allow outgoing communication on UDP port 53 for Synapse Studio. To connect using tools such as SSMS and Power BI, you must allow outgoing communication on TCP port 1433.

https://docs.microsoft.com/en-us/azure/synapse-analytics/security/synapse-workspace-ip-firewall

Virtual networks

Azure Virtual Network (VNet) enables private networks in Azure. VNet enables many types of Azure resources, such as Azure Synapse Analytics, to securely communicate with other virtual networks, the internet, and on-premises networks. When you create your Azure Synapse workspace, you can choose to associate it to a Microsoft Azure Virtual Network. The Virtual Network associated with your workspace is managed by Azure Synapse. This Virtual Network is called a Managed workspace Virtual Network.

Using a managed workspace virtual network provides the following benefits:

• With a Managed workspace Virtual Network, you can offload the burden of managing the Virtual Network to Azure Synapse.

• You don't have to configure inbound NSG rules on your own Virtual Networks to allow Azure Synapse management traffic to enter your Virtual Network. Misconfiguration of these NSG rules causes service disruption for customers.

• You don't need to create a subnet for your Spark clusters based on peak load.

• Managed workspace Virtual Network along with Managed private endpoints protects against data exfiltration. You can only create Managed private endpoints in a workspace that has a Managed workspace Virtual Network associated with it.

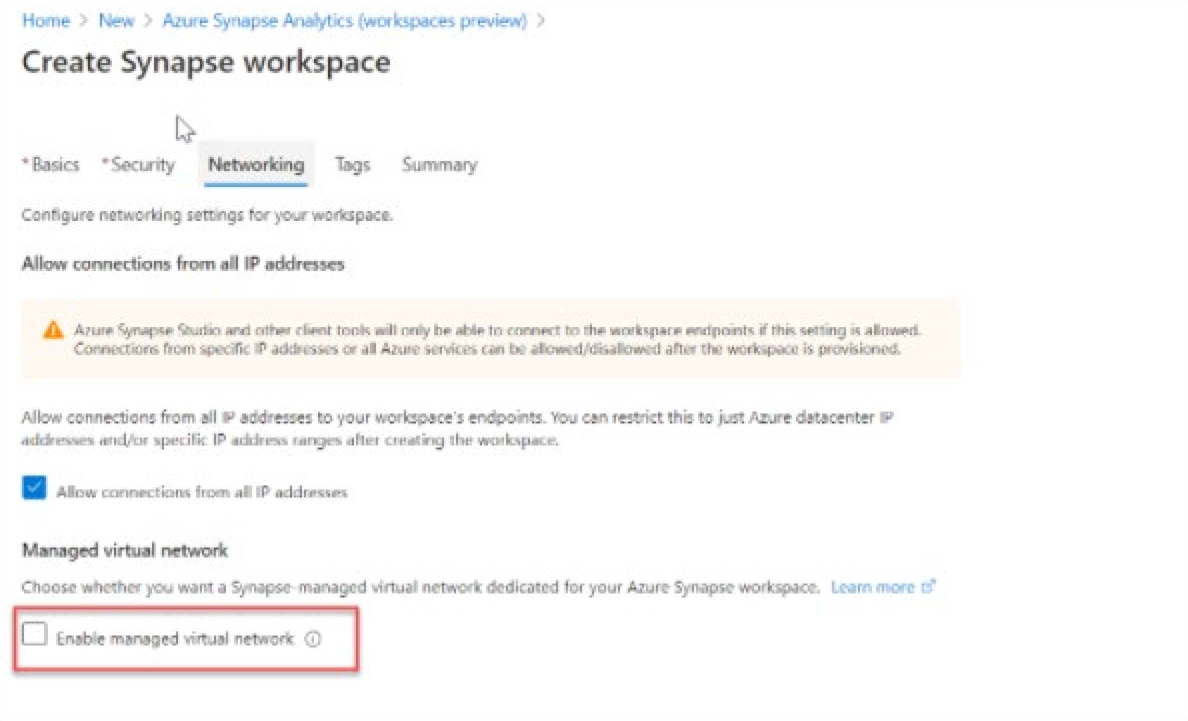
• It ensures that your workspace is network isolated from other workspaces.

If your workspace has a Managed workspace Virtual Network, Data integration and Spark resources are deployed in it. A Managed workspace Virtual Network also provides user-level isolation for Spark activities because each Spark cluster is in its own subnet.

Dedicated SQL pool and serverless SQL pool are multi-tenant capabilities and therefore reside outside of the Managed workspace Virtual Network. Intra-workspace communication to dedicated SQL pool and serverless SQL pool use Azure private links.

These private links are automatically created for you when you create a workspace with a Managed workspace Virtual Network associated to it.

You can only choose to enable managed virtual networks as you are creating the Azure Synapse Workspaces.



Home > New > Azure Synapse Analytics (workspaces preview) >

Create Synapse workspace

*Basics *Security Networking Tags Summary

Configure networking settings for your workspace.

Allow connections from all IP addresses

Azure Synapse Studio and other client tools will only be able to connect to the workspace endpoints if this setting is allowed. Connections from specific IP addresses or all Azure services can be allowed/disallowed after the workspace is provisioned.

Allow connections from all IP addresses to your workspace's endpoints. You can restrict this to just Azure datacenter IP addresses and/or specific IP address ranges after creating the workspace.

☑ Allow connections from all IP addresses

Managed virtual network

Choose whether you want a Synapse-managed virtual network dedicated for your Azure Synapse workspace. Learn more

☐ Enable managed virtual network

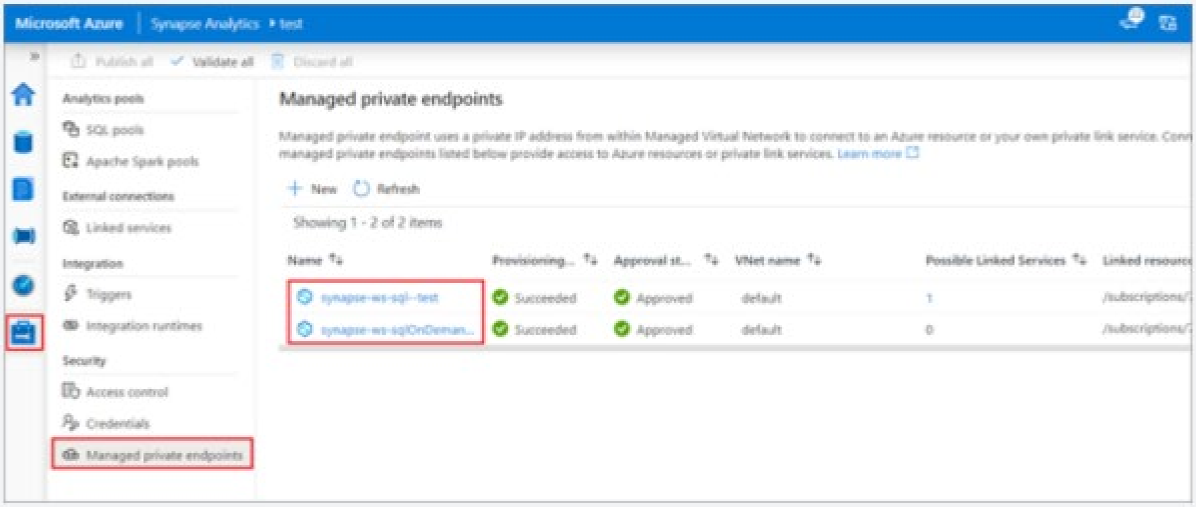
https://docs.microsoft.com/en-us/azure/synapse-analytics/security/synapse-workspace-managed-vnet

Private endpoints

Azure Synapse Analytics enables you to connect upto its various components through endpoints. You can set up managed private endpoints to access these components in a secure manner known as private links. This can only be achieved in an Azure Synapse workspace with a Managed workspace Virtual Network. Private link enables you to access Azure services (such as Azure Storage and Azure Cosmos DB) and Azure hosted customer/partner services from your Azure Virtual Network securely.

When you use a private link, traffic between your Virtual Network and workspace traverses entirely over the Microsoft backbone network. Private Link protects against data exfiltration risks. You establish a private link to a resource by creating a private endpoint.

Private endpoint uses a private IP address from your Virtual Network to effectively bring the service into your Virtual Network. Private endpoints are mapped to a specific resource in Azure and not the entire service. Customers can limit connectivity to a specific resource approved by their organization. You can manage the private endpoints in the Azure Synapse Studio manage hub.



Managed private endpoints

Managed private endpoint uses a private IP address from within Managed Virtual Network to connect to an Azure resource or your own private link service. Conn managed private endpoints listed below provide access to Azure resources or private link services. Learn more

Showing 1 - 2 of 2 items

Name	Provisioning...	Approval st...	VNet name	Possible Linked Services	Linked resourc
synapse-ws-sql--test	Succeeded	Approved	default	1	/subscriptions/
synapse-ws-sqlOnDeman...	Succeeded	Approved	default	0	/subscriptions/

https://docs.microsoft.com/en-us/azure/synapse-analytics/security/synapse-workspace-managed-private-endpoints

Question 77: Skipped
Azure HDInsight is a low-cost cloud solution which provides technologies to help you ingest, process, and analyze big data.

Which of the following are supported in the HDInsight solution? (Select all that apply)

- □ PowerShell
- □ Interactive Query (Correct)
- □ Kafka (Correct)
- □ Sentinel
- □ Spark (Correct)
- □ Sphere
- □ Storm (Correct)
- □ Repos
- □ Hadoop (Correct)
- □ Hbase (Correct)

Explanation
Azure HDInsight provides technologies to help you ingest, process, and analyze big data. It supports batch processing, data warehousing, IoT, and data science.

Key features

HDInsight is a low-cost cloud solution. HDInsight supports the latest open-source projects from the Apache Hadoop and Spark ecosystems. It includes Apache Hadoop, Spark, Kafka, HBase, Storm, and Interactive Query.

• **Hadoop** includes Apache Hive, HBase, Spark, and Kafka. Hadoop stores data in a file system (HDFS). Spark stores data in memory. This difference in storage makes Spark about 100 times faster.

• **HBase** is a NoSQL database built on Hadoop. It's commonly used for search engines. HBase offers automatic failover.

• **Storm** is a distributed real-time streamlining analytics solution.

• **Kafka** is an open-source platform that's used to compose data pipelines. It offers message queue functionality, which allows users to publish or subscribe to real-time data streams.

Ingesting data

As a data engineer, use Hive to run ETL operations on the data you're ingesting. Or orchestrate Hive queries in Azure Data Factory.

https://azure.microsoft.com/en-us/services/hdinsight/#features

Question 78: **Skipped**

What function provides a `rowset` view over a JSON document?

- ○ `VIEWRSET`
- ○ `OPENROWSET`
- ○ `WITH`
- ○ `OPENJSON` **(Correct)**

Explanation

`OPENJSON` (Transact-SQL) is a table-valued function that parses JSON text and returns objects and properties from the JSON input as rows and columns. In other words, `OPENJSON` provides a rowset view over a JSON document. You can explicitly specify the columns in the rowset and the JSON property paths used to populate the columns. Since `OPENJSON` returns a set of rows, you can use `OPENJSON` in the FROM clause of a Transact-SQL statement just as you can use any other table, view, or table-valued function.

Use `OPENJSON` to import JSON data into SQL Server, or to convert JSON data to relational format for an app or service that can't consume JSON directly.

The `OPENJSON` function provides a `rowset` view over a JSON document.

https://docs.microsoft.com/en-us/sql/t-sql/functions/openjson-transact-sql?view=sql-server-ver15

Question 79: Skipped

True or False: Azure Blob Storage is the least expensive method to store data and one of it best features is that it allows for querying the data directly within the Blob environment.

- ○ False (Correct)
- ○ True

Explanation

Azure Blob Storage Queries

If you create a storage account as a Blob store, you can't query the data directly. To query it, either move the data to a store that supports queries or set up the Azure Storage account for a data lake storage account. Azure Blob storage has no API to query data within the blob - it's just dumb storage. You're essentially limited to reading, deserializing and grabbing your value(s).

https://stackoverflow.com/questions/38721458/query-blobs-in-blob-storage

---

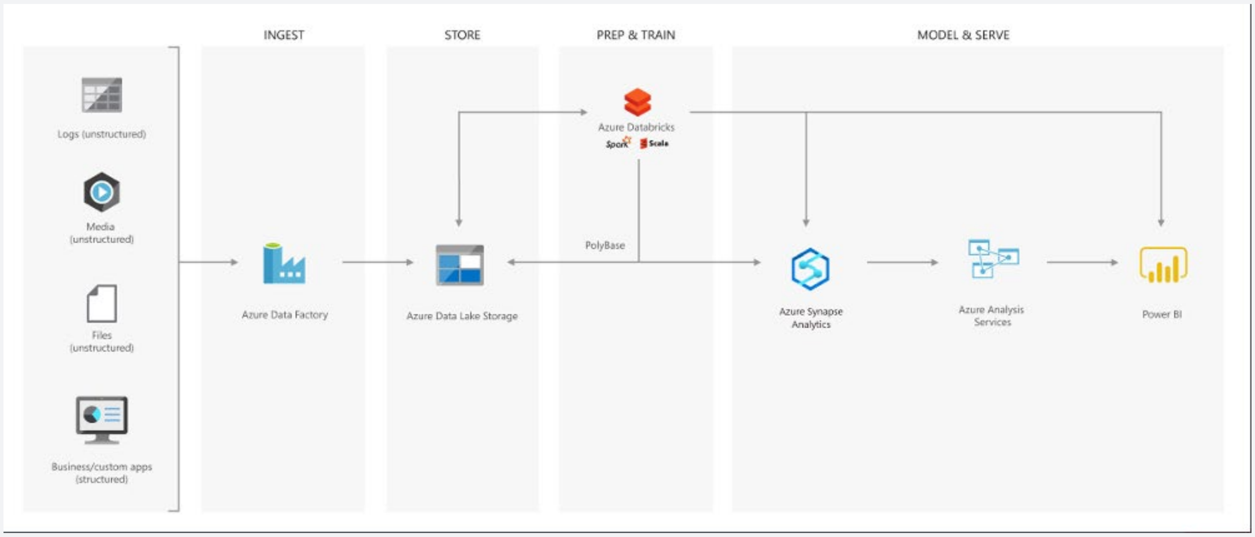
Question 80: **Skipped**

Scenario: You are a Data Engineering consultant for a Avengers Security. In the past, they've created an on-premises business intelligence solution that used a Microsoft SQL Server Database Engine, SQL Server Integration Services, SQL Server Analysis Services, and SQL Server Reporting Services to provide historical reports. They tried using the Analysis Services Data Mining component to create a predictive analytics solution to predict the buying behaviour of customers. While this approach worked well with low volumes of data, it couldn't scale after more than a gigabyte of data was collected. Furthermore, they were never able to deal with the JSON data that a third-party application generated when a customer used the feedback module of the point of sale (POS) application.

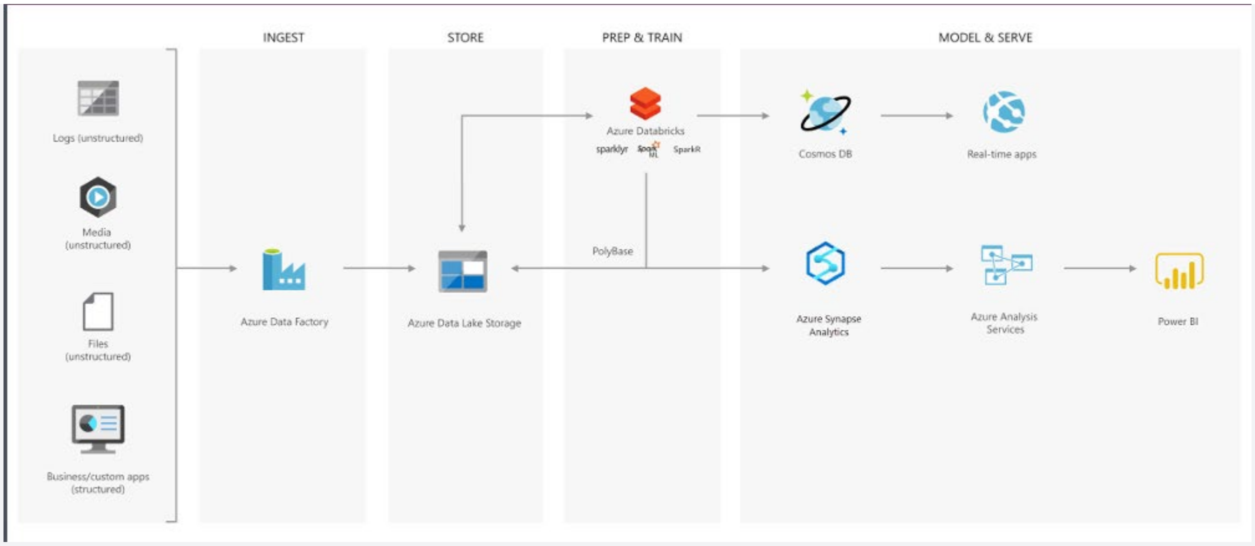
The company has turned to you for help with creating an architecture that can scale with the data needs that are required to create a predictive model and to handle the JSON data so that it's integrated into the BI solution.

Review the following architecture designs.

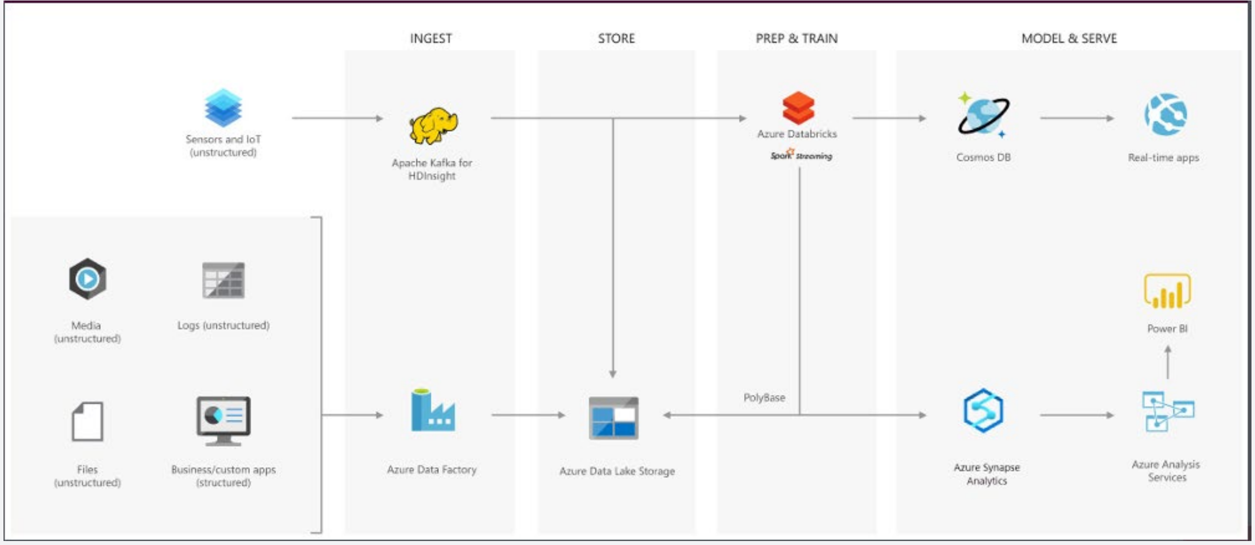
Design A:



Design B:



Design C:



Which architecture would be best suited for the need?

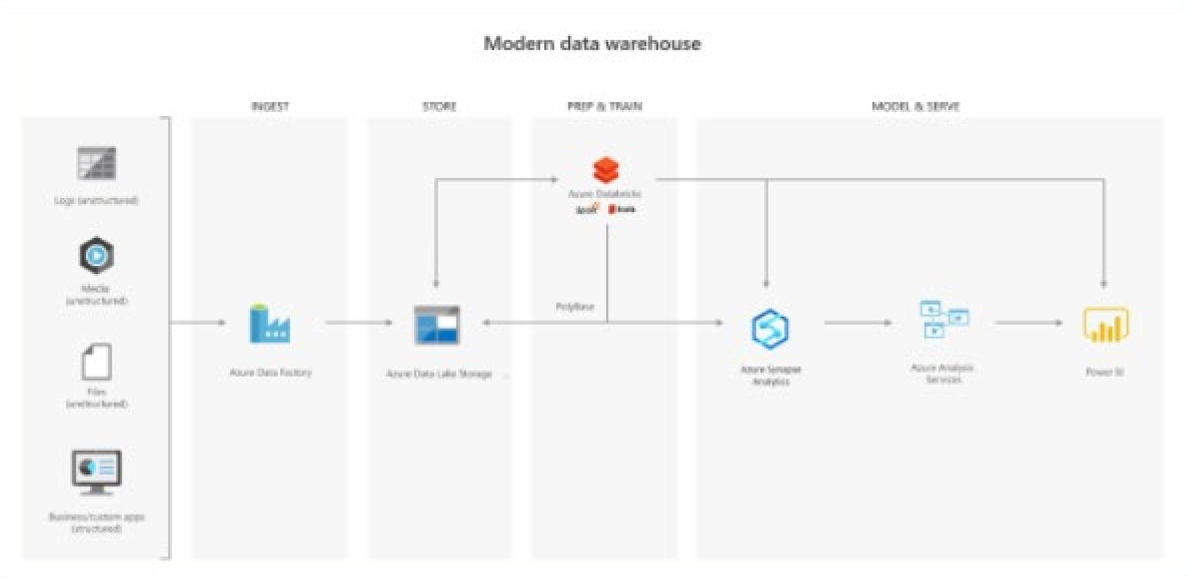
- ○ Design A
  (Correct)
- ○ Design B
- ○ None of the listed options
- ○

Design C

Explanation
Creating a modern data warehouse

Imagine you're a Data Engineering consultant for a Avengers Security. In the past, they've created an on-premises business intelligence solution that used a Microsoft SQL Server Database Engine, SQL Server Integration Services, SQL Server Analysis Services, and SQL Server Reporting Services to provide historical reports. They tried using the Analysis Services Data Mining component to create a predictive analytics solution to predict the buying behaviour of customers. While this approach worked well with low volumes of data, it couldn't scale after more than a gigabyte of data was collected. Furthermore, they were never able to deal with the JSON data that a third-party application generated when a customer used the feedback module of the point of sale (POS) application.

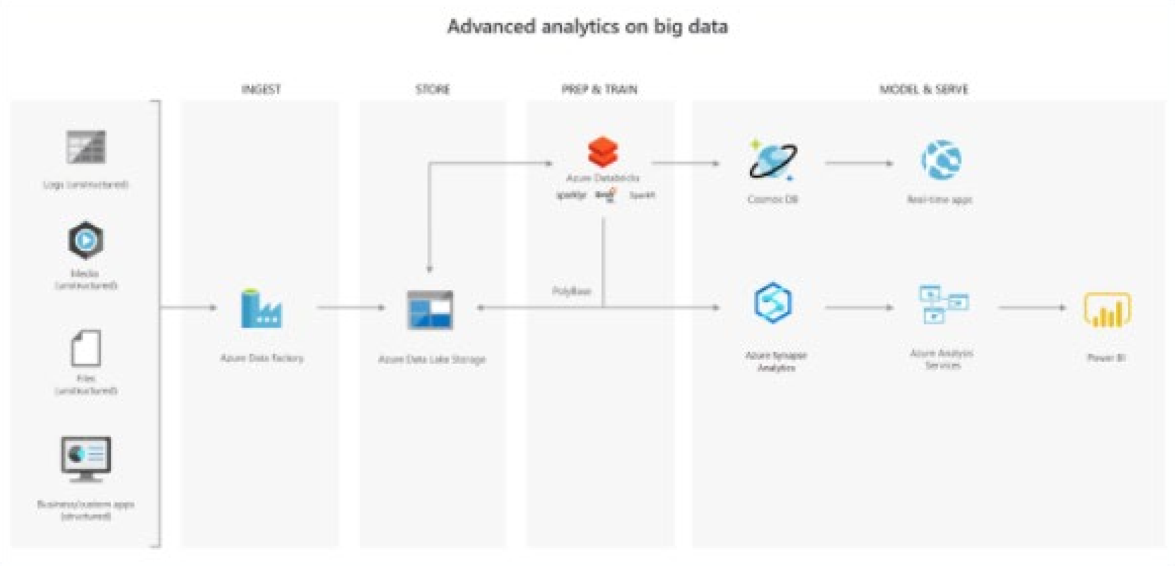
The company has turned to you for help with creating an architecture that can scale with the data needs that are required to create a predictive model and to handle the JSON data so that it's integrated into the BI solution. You suggest the following architecture:



The architecture uses Azure Data Lake Storage at the centre of the solution for a modern data warehouse. Integration Services is replaced by Azure Data Factory to ingest data into the Data Lake from a business application. This is the source for the predictive model that is built into Azure Databricks. PolyBase is used to transfer the historical data into a big data relational format that is held in Azure Synapse Analytics, which also stores the results of the trained model from Databricks. Azure Analysis Services provides the caching capability for SQL Data Warehouse to service many users and to present the data through Power BI reports.

Advanced analytics for big data

In this second use case, Azure Data Lake Storage plays an important role in providing a large-scale data store. Your skills are needed by Hydra Corporation, which is a global seller of bicycles and cycling components through a chain of resellers and on the internet. As their customers browse the product catalogue on their websites and add items to their baskets, a recommendation engine that is built into Azure Databricks recommends other products. They need to make sure that the results of their recommendation engine can scale globally. The recommendations are based on the web log files that are stored on the web servers and transferred to the Azure Databricks model hourly. The response time for the recommendation should be less than 1 ms. You propose the following architecture:



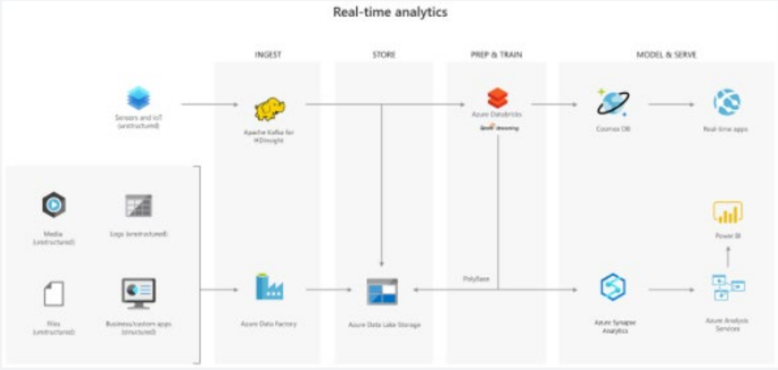
Real-time analytical solutions

To perform real-time analytical solutions, the ingestion phase of the architecture is changed for processing big data solutions. In this architecture, note the introduction of Apache Kafka for Azure HDInsight to ingest streaming data from an Internet of Things (IoT) device, although this could be replaced with Azure IoT Hub and Azure Stream

Analytics. The key point is that the data is persisted in Data Lake Storage Gen2 to service other parts of the solution.

In this use case, you are a Data Engineer for HAMMER Industries, an organization that is working with a transport company to monitor the fleet of Heavy Goods Vehicles (HGV) that drive around Europe. Each HGV is equipped with sensor hardware that will continuously report metric data on the temperature, the speed, and the oil and brake solution levels of an HGV. When the engine is turned off, the sensor also outputs a file with summary information about a trip, including the mileage and elevation of a trip. A trip is a period in which the HGV engine is turned on and off.

Both the real-time data and batch data is processed in a machine learning model to predict a maintenance schedule for each of the HGVs. This data is made available to the downstream application that third-party garage companies can use if an HGV breaks down anywhere in Europe. In addition, historical reports about the HGV should be visually presented to users. As a result, the following architecture is proposed:



In this architecture, there are two ingestion streams. Azure Data Factory ingests the summary files that are generated when the HGV engine is turned off. Apache Kafka provides the real-time ingestion engine for the telemetry data. Both data streams are stored in Azure Data Lake Store for use in the future, but they are also passed on to other technologies to meet business needs. Both streaming and batch data are provided to the predictive model in Azure Databricks, and the results are published to Azure Cosmos DB to be used by the third-party garages. PolyBase transfers data from the Data Lake Store into SQL Data Warehouse where Azure Analysis Services creates the HGV reports by using Power BI.

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

Question 81: Skipped

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

When a table is created, by default the data structure has no indexes and is called a(n) [?].

- ○ NoMap object
- ○ Heap (Correct)
- ○ N-tree
- ○ Open table

Explanation

When a table is created, by default the data structure has no indexes and is called a heap. A well-designed indexing strategy can reduce disk I/O operations and consume less system resources therefore improving query performance, especially when using filtering, scans, and joins in a query.

Dedicated SQL Pools have the following indexing options available:

Clustered columnstore index

Dedicated SQL Pools create a clustered columnstore index when no index options are specified on a table. Clustered columnstore indexes offer both the highest level of data compression as well as the best overall query performance. Clustered columnstore indexes will generally outperform clustered rowstore indexes or heap tables and are usually the best choice for large tables.

Additional compression on the data can be gained also with the index option COLUMNSTORE_ARCHIVE. These reduced sizes allow less memory to be used when accessing and using the data as well as reducing the IOPs required to retrieve data from storage.

Columnstore works on segments of 1,024,000 rows that get compressed and optimized by column. This segmentation further helps to filter out and reduce the data accessed through leveraging metadata stored which summarizes the range and values within each segment during query optimization.

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

Clustered index

Clustered Rowstore Indexes define how the table itself is stored, ordered by the columns used for the Index. There can be only one clustered index on a table.

Clustered indexes are best for queries and joins that require ranges of data to be scanned, preferably in the same order that the index is defined.

Non-clustered index

A non-clustered index can be defined on a table or view with a clustered index or on a heap. Each index row in the non-clustered index contains the non-clustered key value and a row locator. This is a data structure separate/additional to the table or heap. You can create multiple non-clustered indexes on a table.

Non clustered indexes are best used when used for the columns in a join, group by statement or where clauses that return an exact match or few rows.

https://docs.microsoft.com/en-us/sql/t-sql/statements/create-table-azure-sql-data-warehouse?view=aps-pdw-2016-au7

Question 82: Skipped

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

Within Azure Synapse SQL, [?] stores a copy of the result set on the control node so that queries do not need to pull data from the storage subsystem or compute nodes.

- ○ Result-set caching (Correct)
- ○ Site caching
- ○ Server caching
- ○ VM caching
- ○ Browser caching

Explanation

Enable result-set caching when you expect results from queries to return the same values.

This option stores a copy of the result set on the control node so that queries do not need to pull data from the storage subsystem or compute nodes. The capacity for the result set cache is 1 TB and the data within the result-set cache is expired and purged after 48 hours of not being accessed.

Azure Synapse SQL automatically caches query results in the user database for repetitive use. Result-set caching allows subsequent query executions to get results directly from the persisted cache so recomputation is not needed. Result set caching improves query performance and reduces compute resource usage.

To enable result set caching, run this command when connecting to the MASTER database.

```
SQL

ALTER DATABASE [database_name]

SET RESULT_SET_CACHING ON;
```

https://docs.microsoft.com/en-us/azure/synapse-analytics/sql-data-warehouse/performance-tuning-result-set-caching

Question 83: Skipped

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

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 boundary which represents a process bottleneck which Spark will break this one job into two stages.

In **Stage #1**, Spark will create a pipeline of transformations in which the data is read into RAM.

For **Stage #2**, Spark will again create a pipeline of transformations in which the shuffle data is read into RAM.

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).

What is the main benefit of working backward through your action's lineage?

- ○ It allows Azure to distribute the load to the required number of processors to optimize the load.
- ○ It allows Spark to work on various activities simultaneously using multiple nodes.
- ○ It serializes the work to make the work sequential, thereby lowering CPU and RAM cost.
- ○ It allows Spark to determine if it is necessary to execute every transformation. **(Correct)**

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

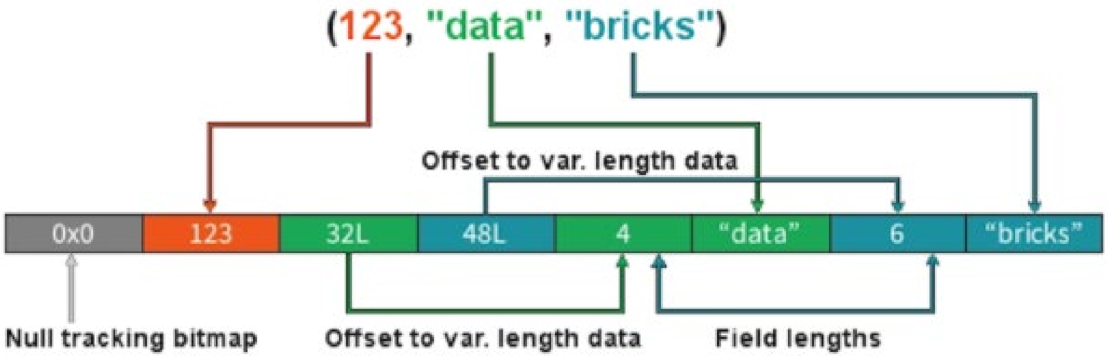
- The first field, "123", is stored in place as its primitive.

- 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

Question 84: **Skipped**

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

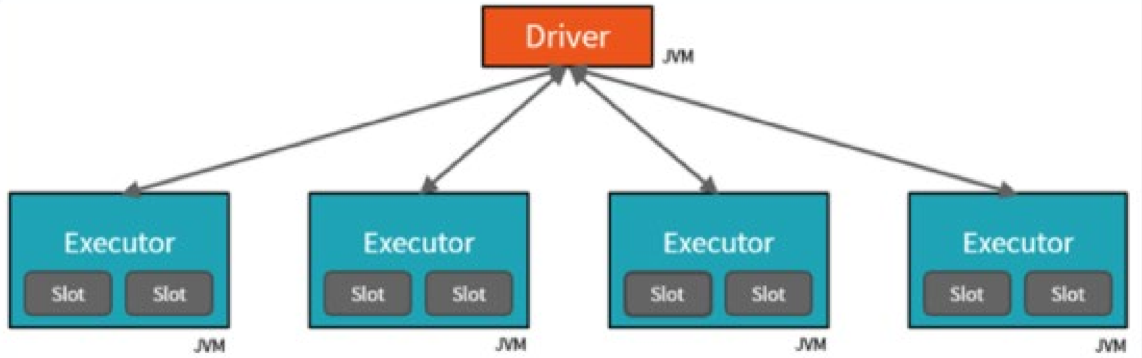
Spark is a Distributed computing environment. The unit of distribution is a Spark Cluster. Every Cluster has a Driver and one or more executors. Work submitted to the Cluster ... [?]

- ○ split into as many independent Jobs as needed.
  (Correct)
- ○ must decide how to partition the data so that it can be distributed for parallel processing.
- ○ specifies the types and sizes of the virtual machines.
- ○ divided into a maximum of 10 independent Jobs.

Explanation

Spark is a Distributed computing environment. 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 as many independent Jobs as needed. This is how work is distributed across the Cluster's nodes. Jobs are further subdivided into tasks. The input to a job is partitioned into one or more partitions. These partitions are the unit of work for each slot. In between tasks, partitions may need to be re-organized and shared over the network.

The cluster: Drivers, executors, slots & tasks



- The **Driver** is the JVM in which our application runs.

- The secret to Spark's awesome performance is parallelism.

    - Scaling vertically is limited to a finite amount of RAM, Threads and CPU speeds.

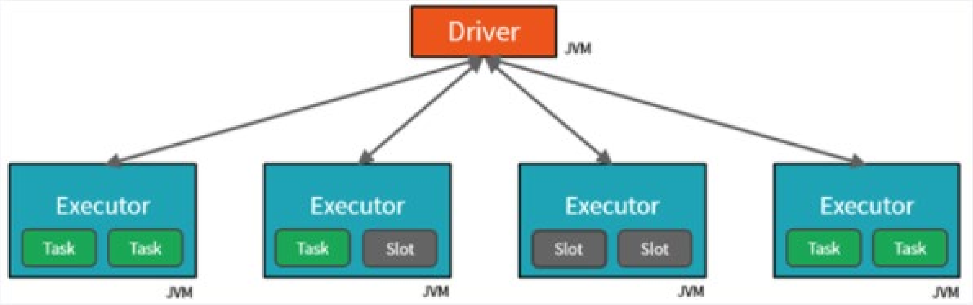
    - Scaling horizontally means we can simply add new "nodes" to the cluster almost endlessly.

- We parallelize at two levels:

    - The first level of parallelization is the **Executor** - a Java virtual machine running on a node, typically, one instance per node.

    - The second level of parallelization is the **Slot** - the number of which is determined by the number of cores and CPUs of each node.

- Each **Executor** has a number of **Slots** to which parallelized **Tasks** can be assigned to it by the **Driver**.



- The JVM is naturally multithreaded, but a single JVM, such as our **Driver**, has a finite upper limit.

- By creating **Tasks**, the **Driver** can assign units of work to **Slots** for parallel execution.

- Additionally, the **Driver** must also decide how to partition the data so that it can be distributed for parallel processing (not shown here).

- Consequently, the **Driver** is assigning a **Partition** of data to each task - in this way each **Task** knows which piece of data it is to process.

- Once started, each **Task** will fetch from the original data source the **Partition** of data assigned to it.

Jobs & stages

- Each parallelized action is referred to as a **Job**.
- The results of each **Job** (parallelized/distributed action) is returned to the **Driver**.
- Depending on the work required, multiple **Jobs** will be required.
- Each **Job** is broken down into **Stages**.
  - This would be analogous to building a house (the job)
  - The first stage would be to lay the foundation.
  - The second stage would be to erect the walls.
  - The third stage would be to add the room.
  - Attempting to do any of these steps out of order just won't make sense, if not just impossible.

Cluster management

- At a much lower level, Spark Core employs a **Cluster Manager** that is responsible for provisioning nodes in our cluster.
  - Databricks provides a robust, high-performing **Cluster Manager** as part of its overall offerings.
- In each of these scenarios, the **Driver** is [presumably] running on one node, with each **Executors** running on N different nodes.
- From a developer's and learner's perspective my primary focus is on...
  - The number of **Partitions** my data is divided into.
  - The number of **Slots** I have for parallel execution.
  - How many **Jobs** am I triggering?
  - And lastly the **Stages** those jobs are divided into.

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

Question 85: Skipped

All data written to Azure Storage is automatically encrypted by Storage Service Encryption (SSE) with a 256-bit Advanced Encryption Standard (AES) cipher, and is FIPS 140-2 compliant.

True or False: For virtual machines (VMs), Azure lets you encrypt virtual hard disks (VHDs) by using Azure Disk Encryption. If someone gets access to the VHD image and downloads it, they can't access the data on the VHD unless they have an Azure Storage account as well. If a bad actor restores the image within their own Azure environment, they will have access to the data on the image.

- ○ True
- ○ False (Correct)

Explanation

Encryption at rest

All data written to Azure Storage is automatically encrypted by Storage Service Encryption (SSE) with a 256-bit Advanced Encryption Standard (AES) cipher, and is FIPS 140-2 compliant. SSE automatically encrypts data when writing it to Azure Storage. When you read data from Azure Storage, Azure Storage decrypts the data before returning it. This process incurs no additional charges and doesn't degrade performance. It can't be disabled.

For virtual machines (VMs), Azure lets you encrypt virtual hard disks (VHDs) by using Azure Disk Encryption. This encryption uses BitLocker for Windows images, and it uses dm-crypt for Linux.

Azure Key Vault stores the keys automatically to help you control and manage the disk-encryption keys and secrets. So even if someone gets access to the VHD image and downloads it, they can't access the data on the VHD.

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

Question 86: **Skipped**

Which Workload Management capability manages minimum and maximum resource allocations during peak periods?

- ○ Workload Isolation **(Correct)**
- ○ Workload Classification
- ○ Workload Importance
- ○ Workload Containment

Explanation

Workload Isolation assigns maximum and minimum usage values for varying resources under load. These adjustments can be done live without having to take the SQL Pool offline.

Dedicated SQL pool workload management in Azure Synapse consists of three high-level concepts:

• Workload Classification

• Workload Importance

• Workload Isolation

Workload isolation

Workload isolation reserves resources for a workload group. Resources reserved in a workload group are held exclusively for that workload group to ensure execution. Workload groups also allow you to define the amount of resources that are assigned per request, much like resource classes do. Workload groups give you the ability to reserve or cap the amount of resources a set of requests can consume. Finally, workload groups are a mechanism to apply rules, such as query timeout, to requests.

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

Question 87: Skipped

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

Azure Monitor provides base-level infrastructure metrics and logs for most Azure services. Azure diagnostic logs are emitted by a resource and provide rich, frequent data about the operation of that resource. Azure Data Factory (ADF) can write diagnostic logs in Azure Monitor.

Data Factory stores pipeline-run data for [?] days.

- ○ 21
- ○ 15
- ○ 10
- ○ 45 (Correct)
- ○ 30

Explanation
Monitor using Azure Monitor

Azure Monitor provides base-level infrastructure metrics and logs for most Azure services. Azure diagnostic logs are emitted by a resource and provide rich, frequent data about the operation of that resource. Azure Data Factory (ADF) can write diagnostic logs in Azure Monitor.

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

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

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

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

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

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

You can also use a storage account or event-hub namespace that isn't in the subscription of the resource that emits logs. The user who configures the setting must have appropriate Azure role-based access control (Azure RBAC) access to both subscriptions.

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

Question 88: Skipped

Scenario: Big Belly Foods, Inc. (BB) owns and operates 300 convenience stores across LatAm. The company sells a variety of packaged foods and drinks, as well as a variety of prepared foods, such as sandwiches and pizzas. The company has a loyalty club whereby members can get daily discounts on specific items by providing their membership number at checkout.

BB employs business analysts who prefer to analyze data by using Microsoft Power BI, and data scientists who prefer analyzing data in Azure Databricks notebooks. You have been hired as an Azure Expert SME and you are to consult the IT team on various Azure related projects.

Business Requirements:

BB wants to create a new analytics environment in Azure to meet the following requirements:

• See inventory levels across the stores. Data must be updated as close to real time as possible.

• Execute ad hoc analytical queries on historical data to identify whether the loyalty club discounts increase sales of the discounted products.

• Every four hours, notify store employees about how many prepared food items to produce based on historical demand from the sales data.

Technical Requirements:

BB identifies the following technical requirements:

• Minimize the number of different Azure services needed to achieve the business goals.

• Use platform as a service (PaaS) offerings whenever possible and avoid having to provision virtual machines that must be managed by BB.

• Ensure that the analytical data store is accessible only to the company's on-premises network and Azure services.

• Use Azure Active Directory (Azure AD) authentication whenever possible.

• Use the principle of least privilege when designing security.

• Stage Inventory data in Azure Data Lake Storage Gen2 before loading the data into the analytical data store. BB wants to remove transient data from Data

• Lake Storage once the data is no longer in use. Files that have a modified date that is older than 14 days must be removed.

• Limit the business analysts' access to customer contact information, such as phone numbers, because this type of data is not analytically relevant.

• Ensure that you can quickly restore a copy of the analytical data store within one hour in the event of corruption or accidental deletion.

Planned Environment:

BB plans to implement the following environment:

• The application development team will create an Azure event hub to receive real-time sales data, including store number, date, time, product ID, customer loyalty number, price, and discount amount, from the point of sale (POS) system and output the data to data storage in Azure.

• Customer data, including name, contact information, and loyalty number, comes from Salesforce, a SaaS application, and can be imported into Azure once every eight hours. Row modified dates are not trusted in the source table.

• Product data, including product ID, name, and category, comes from Salesforce and can be imported into Azure once every eight hours. Row modified dates are not trusted in the source table.

• Daily inventory data comes from a Microsoft SQL server located on a private network.

• BB currently has 5 TB of historical sales data and 100 GB of customer data. The company expects approximately 100 GB of new data per month for the next year.

• BB will build a custom application named FoodPrep to provide store employees with the calculation results of how many prepared food items to produce every four hours.

• BB does not plan to implement Azure ExpressRoute or a VPN between the on-premises network and Azure.

The Ask:

The team looks to you for direction on what should be done to improve high availability of the real-time data processing solution. Which of the following should you propose as the best solution?

- ○ Set Data Lake Storage to use geo-redundant storage (GRS).

- ○ Deploy identical Azure Stream Analytics jobs to paired regions in Azure. **(Correct)**
- ○ Deploy an Azure Stream Analytics job and use an Azure Automation runbook to check the status of the job and to start the job if it stops.
- ○ Deploy a High Concurrency Databricks cluster.

Explanation

The best solution to move forward is to deploy identical Azure Stream Analytics jobs to paired regions in Azure. The application development team should create an Azure event hub to receive real-time sales data, including store number, date, time, product ID, customer loyalty number, price, and discount amount, from the point of sale (POS) system and output the data to data storage in Azure.

Guarantee Stream Analytics job reliability during service updates

Part of being a fully managed service is the capability to introduce new service functionality and improvements at a rapid pace. As a result, Stream Analytics can have a service update deploy on a weekly (or more frequent) basis. No matter how much testing is done there is still a risk that an existing, running job may break due to the introduction of a bug. If you are running mission critical jobs, these risks need to be avoided. You can reduce this risk by following Azure's **paired region** model.

How do Azure paired regions address this concern?

Stream Analytics guarantees jobs in paired regions are updated in separate batches. As a result there is a sufficient time gap between the updates to identify potential issues and remediate them. *With the exception of Central India* (whose paired region, South India, does not have Stream Analytics presence), the deployment of an update to Stream Analytics would not occur at the same time in a set of paired regions. Deployments in multiple regions **in the same group** may occur **at the same time**.

The article on **availability and paired regions** has the most up-to-date information on which regions are paired. It is recommended to deploy identical jobs to both paired regions. You should then monitor these jobs to get notified when something unexpected happens. If one of these jobs ends up in a Failed state after a Stream Analytics service update, you can contact customer support to help identify the root cause. You should also fail over any downstream consumers to the healthy job output.

https://docs.microsoft.com/en-us/azure/stream-analytics/stream-analytics-job-reliability

Question 89: Skipped

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

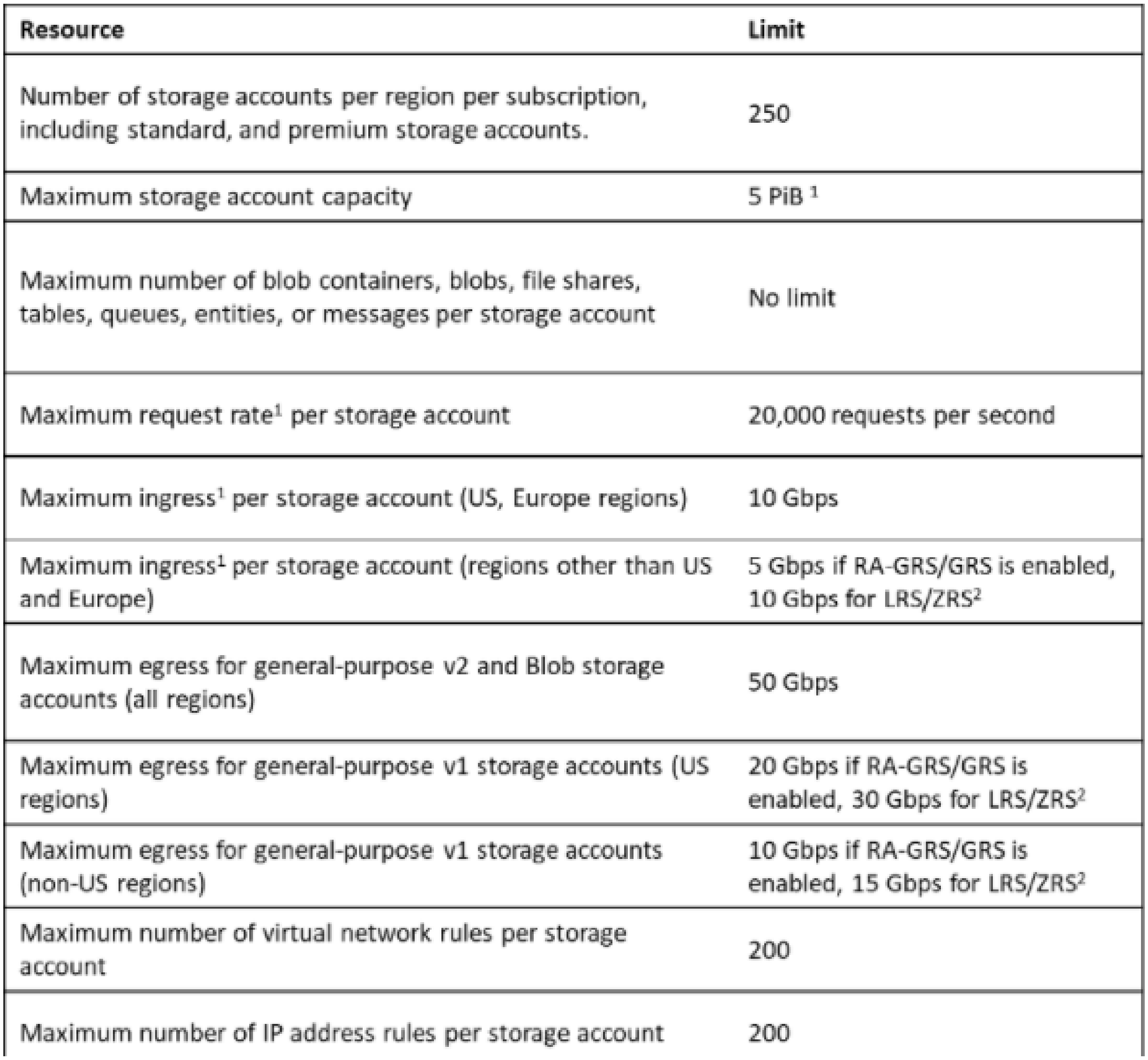
Microsoft Azure Storage is a managed service that provides durable, secure, and scalable storage in the cloud. A single Azure subscription can host up to [A] storage accounts, each of which can hold [B] TB of data.

- ○ [A] 500, [B] 1000
- ○ [A] 200, [B] 500
- ○ [A] 250, [B] 500 (Correct)
- ○ [A] 500, [B] 500

Explanation

Scale targets for standard storage accounts

The following table describes default limits for Azure general-purpose v1, v2, Blob storage, and block blob storage accounts. The *ingress* limit refers to all data that is sent to a storage account. The *egress* limit refers to all data that is received from a storage account.



Resource	Limit
Number of storage accounts per region per subscription, including standard, and premium storage accounts.	250
Maximum storage account capacity	5 PiB [1]
Maximum number of blob containers, blobs, file shares, tables, queues, entities, or messages per storage account	No limit
Maximum request rate[1] per storage account	20,000 requests per second
Maximum ingress[1] per storage account (US, Europe regions)	10 Gbps
Maximum ingress[1] per storage account (regions other than US and Europe)	5 Gbps if RA-GRS/GRS is enabled, 10 Gbps for LRS/ZRS[2]
Maximum egress for general-purpose v2 and Blob storage accounts (all regions)	50 Gbps
Maximum egress for general-purpose v1 storage accounts (US regions)	20 Gbps if RA-GRS/GRS is enabled, 30 Gbps for LRS/ZRS[2]
Maximum egress for general-purpose v1 storage accounts (non-US regions)	10 Gbps if RA-GRS/GRS is enabled, 15 Gbps for LRS/ZRS[2]
Maximum number of virtual network rules per storage account	200
Maximum number of IP address rules per storage account	200

https://docs.microsoft.com/en-us/azure/storage/common/scalability-targets-standard-account

Question 90: Skipped

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

Security administrators can control data access by using [?] within Data Lake Storage. Built-in security groups include `ReadOnlyUsers`, `WriteAccessUsers`, and `FullAccessUsers`.

- ○ AD OAuth
- ○ AD Desired State Configuration (ADDSC)
- ○ Active Directory Security GroupActive Directory Security Groupss (Correct)
- ○ Active Directory Application Groups

Explanation

Data Lake Storage Data Security

Because Data Lake Storage supports Azure Active Directory ACLs, security administrators can control data access by using the familiar Active Directory Security Groups. Role-based access control (RBAC) is available both in Gen1 and Gen2. Built-in security groups include `ReadOnlyUsers`, `WriteAccessUsers`, and `FullAccessUsers`.

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