1) What is PySpark? / What do you know about PySpark?

PySpark is a tool or interface of Apache Spark developed by the Apache Spark community and Python to support Python to work with Spark. This tool collaborates with Apache Spark using APIs written in Python to support features like Spark SQL, Spark DataFrame, Spark Streaming, Spark Core, Spark MLlib, etc. It provides an interactive PySpark shell to analyze structured and semi-structured data in a distributed environment and process them by providing optimized APIs that help the program to read data from various data sources. PySpark features are implemented in the py4j library in Python. Due to the availability of the Py4j library, it facilitates users to work with RDDs (Resilient Distributed Datasets) in the Python programming language. Python supports many libraries that support big data processing and machine learning.

You can install PySpark using PyPi by using the following command:

1. pip install pyspark

2) What are the main characteristics of PySpark?

Following are the main four main characteristics of PySpark:

* **Nodes are abstracted:** The nodes are abstracted in PySpark. It means we cannot access the individual worker nodes.
* **PySpark is based on MapReduce:** PySpark is based on the MapReduce model of Hadoop. It means that the programmer provides the map and the reduced functions.
* **APIs for Spark features:** PySpark provides APIs for utilizing Spark features.
* **Abstracted Network:** PySpark provides abstracted networks. It means that the networks are abstracted in PySpark, and it facilitates only implicit communication.

3) What is RDD in PySpark?

In PySpark, RDD is an acronym that stands for Resilient Distributed Datasets. It is a core data structure of PySpark. It is a low-level object that is highly efficient in performing distributed tasks.

The PySpark's RDDs are the elements that can run and operate on multiple nodes to do parallel processing on a cluster. These are immutable elements. It means that if you once create an RDD, you cannot change it. RDDs are also fault-tolerant. In the case of any failure, they recover automatically. We can apply multiple operations on RDDs to achieve a certain task.

4) What are the key advantages and disadvantages of PySpark?

Following is a list of key advantages and disadvantages of PySpark:

**Advantages of PySpark**

* PySpark is an easy-to-learn language. You can learn and implement it easily if you know Python and Apache Spark.
* PySpark is simple to use. It provides parallelized codes that are simple to write.
* Error handling is simple in the PySpark framework. You can easily handle errors and manage synchronization points
* PySpark is a Python API for Apache Spark. It provides great library support. Python has a huge library collection for working in data science and data visualization compared to other languages.
* Many important algorithms are already written and implemented in Spark. It provides many algorithms in Machine Learning or Graphs.

**Disadvantages of PySpark**

* PySpark is based on Hadoop's MapReduce model, so sometimes, it becomes challenging to manage and express problems using the MapReduce model.
* Since Apache Spark was originally written in Scala while using PySpark in Python programs, they are not as efficient as other programming models. It is approximate 10x times slower than the Scala programs. Due to this reason, it negatively impacts the performance of heavy data processing applications.
* The Spark Streaming API in PySpark is not as efficient as Scala. It still requires improvements.
* In PySpark, the nodes are abstracted, and it uses the abstracted network, so it cannot be used to modify the internal function of the Spark. Scala is preferred in this case.

5) What are the prerequisites to learn PySpark?

PySpark is easy to learn and implement. It doesn't require the expertise of many programming languages or databases. You can learn it easily if you know a programming language and framework. Before learning the concept of PySpark, you should learn some knowledge of Apache Spark and Python. It will be very helpful to learn the advanced concepts of PySpark.

6) Why are Partitions immutable in PySpark?

In PySpark, every transformation generates a new partition. Partitions use HDFS API to make partitions immutable, distributed, and fault-tolerant. Partitions are also aware of data locality.

7) What are the key differences between an RDD, a DataFrame, and a DataSet?

Following are the key differences between an RDD, a DataFrame, and a DataSet:

**RDD**

* RDD is an acronym that stands for Resilient Distributed Dataset. It is a core data structure of PySpark.
* RDD is a low-level object that is highly efficient in performing distributed tasks.
* RDD is best to do low-level transformations, operations, and control on a dataset.
* RDD is mainly used to alter data with functional programming structures than with domain-specific expressions.
* If you have a similar arrangement of data that needs to be calculated again, RDDs can be efficiently reserved.
* RDD contains all datasets and DataFrames in PySpark.

**DataFrame**

* A DataFrame is equivalent to a relational table in Spark SQL. It facilitates the structure like lines and segments to be seen.
* If you are working on Python, it is best to start with DataFrames and then switch to RDDs if you want more flexibility.
* One of the biggest disadvantages of DataFrames is Compile Time Wellbeing. For example, if the information structure is unknown, you cannot control it.

**DataSet**

* A Dataset is a distributed collection of data. It is a subset of DataFrames.
* Dataset is a newly added interface in Spark 1.6 to provide RDD benefits.
* DataSet consists of the best encoding component. It provides time security in an organized manner, unlike information edges.
* DataSet provides a greater level of type safety at compile-time. It can be used if you want typed JVM objects.
* By using DataSet, you can take advantage of Catalyst optimization. You can also use it to benefit from Tungsten's fast code generation.

8) What do you understand by PySpark SparkContext?

SparkContext acts as the entry point to any spark functionality. When the Spark application runs, it starts the driver program, and the main function and SparkContext get initiated. After that, the driver program runs the operations inside the executors on worker nodes. In PySpark, SparkContext is known as PySpark SparkContext. It uses Py4J (library) to launch a JVM and then creates a JavaSparkContext. The PySpark's SparkContext is by default available as 'sc', so it doesn't mean creating a new SparkContext.

9) What is the usage of PySpark StorageLevel?

The PySpark StorageLevel is used to control the storage of RDD. It controls how and where the RDD is stored. PySpark StorageLevel decides if the RDD is stored on the memory, over the disk, or both. It also specifies whether we need to replicate the RDD partitions or serialize the RDD.

**Following is the code for PySpark StorageLevel:**

1. class pyspark.StorageLevel( useDisk, useMemory, useOfHeap, deserialized, replication = 1)

10) What do you understand by data cleaning?

Data cleaning is the process of preparing data by analyzing the data and removing or modifying data if it is incorrect, incomplete, irrelevant, duplicated, or improperly formatted.

11) What is PySpark SparkConf?

PySpark SparkConf is mainly used if we have to set a few configurations and parameters to run a Spark application on the local/cluster. In other words, we can say that PySpark SparkConf is used to provide configurations to run a Spark application.

12) What are the different types of algorithms supported in PySpark?

Different types of algorithms supported in PySpark are:

* spark.mllib
* mllib.regression
* mllib.recommendation
* mllib.clustering
* mllib.classification
* mllib.linalg
* mllib.fpm

13) What is SparkCore, and what are the key functions of SparkCore?

SparkCore is a general execution engine for the Spark platform, including all the functionalities. It offers in-memory computing capabilities to deliver a good speed, a generalized execution model to support various applications, and Java, Scala, and Python APIs that make the development easy.

The main responsibility of SparkCore is to perform all the basic I/O functions, scheduling, monitoring, etc. It is also responsible for fault recovery and effective memory management.

**The key functions of SparkCore are:**

* Perform all the basic I/O functions
* Job scheduling
* Monitoring jobs
* Memory management
* Fault-tolerance
* Interaction with storage systems

Note: It also includes additional libraries that can divide the workloads for streaming, machine learning, and SQL.

14) What do you know about PySpark SparkFiles?

PySpark facilitates users to upload their files using sc.addFile. Here, sc is our default SparkContext. We can also get the path of the working directory using SparkFiles.get. SparkFiles provides the following types of class methods to resolve the path to the files added through SparkContext.addFile():

* get(filename)
* getrootdirectory()

15) What do you know about PySpark serializers?

In PySpark, serialization is a process that is used to conduct performance tuning on Spark. PySpark supports serializers because we have to continuously check the data sent or received over the network to the disk or memory. PySpark supports two types of serializers. They are as follows:

* **PickleSerializer:** This is used to serialize the objects using Python's PickleSerializer using class pyspark.PickleSerializer). This serializer supports almost every Python object.
* **MarshalSerializer:** The MarshalSerializer is used to perform serialization of objects. This can be used by using class pyspark.MarshalSerializer. This serializer is way faster than the PickleSerializer, but it supports only limited types.

16) What is PySpark ArrayType? Give an example to explain it well.

PySpark ArrayType is a collection data type that extends the PySpark's DataType class, which is the superclass for all kinds. The PySpark ArrayType contains only the same types of items. The ArraType() method can also be used to construct an instance of an ArrayType.

It accepts two arguments:

* **valueType:** The valueType should extend the DataType class in PySpark.
* **valueContainsNull:** It is an optional argument. It specifies whether a value can accept null and is set to True by default.

**Example:**

1. from pyspark.sql.types import StringType, ArrayType
2. arrayCol = ArrayType(StringType(),False)

17) What are the most frequently used Spark ecosystems?

The most frequently used Spark ecosystems are:

* Spark SQL for developers. It is also known as Shark.
* Spark Streaming for processing live data streams.
* Graphx for generating and computing graphs.
* MLlib (also known as Machine Learning Algorithms)
* SparkR to promote R programming language in Spark engine.

18) What machine learning API does PySpark provide?

Just like Apache Spark, PySpark also provides a machine learning API known as MLlib. MLlib supports the following types of machine learning algorithms:

* **mllib.classification:** This machine learning API supports different methods for binary or multiclass classification and regression analysis such as Random Forest, Decision Tree, Naive Bayes, etc.
* **mllib.clustering:** This machine learning API solves clustering problems for grouping entities subsets with one another depending on similarity.
* **mllib.fpm:** FPM stands for Frequent Pattern Matching in this machine learning API. This machine learning API is used to mine frequent items, subsequences, or other structures that are used for analyzing large datasets.
* **mllib.linalg:** This machine learning API is used to solve problems on linear algebra.
* **mllib.recommendation:** This machine learning API is used for collaborative filtering and recommender systems.
* **spark.mllib:** This machine learning API is used to support model-based collaborative filtering where small latent factors are identified using the Alternating Least Squares (ALS) algorithm used for predicting missing entries.
* **mllib.regression:** This machine learning API solves problems by using regression algorithms that find relationships and variable dependencies.

19) What is PySpark Partition? How many partitions can you make in PySpark?

PySpark Partition is a method of splitting a large dataset into smaller datasets based on one or more partition keys. It enhances the execution speed as transformations on partitioned data run quicker because each partition's transformations are executed in parallel. PySpark supports both partitioning in memory (DataFrame) and partitioning on disc (File system). When we make a DataFrame from a file or table, PySpark creates the DataFrame in memory with a specific number of divisions based on specified criteria.

It also facilitates us to create a partition on multiple columns using partitionBy() by passing the columns you want to partition as an argument to this method.

**Syntax:**

1. partitionBy(self, \*cols)

In PySpark, it is recommended to have 4x of partitions to the number of cores in the cluster available for application.

20) What do you understand by PySpark DataFrames?

PySpark DataFrames are the distributed collection of well-organized data. These are the same as relational databases tables and are placed into named columns. PySpark DataFrames are better optimized than R or Python programming language because these can be created from different sources like Hive Tables, Structured Data Files, existing RDDs, external databases, etc.

The biggest advantage of PySpark DataFrame is that the data in the PySpark DataFrame is distributed across different machines in the cluster, and the operations performed on this would be run parallel on all the machines. This facilitates handling a large collection of structured or semi-structured data of a range of petabytes.

21) What do you understand by "joins" in PySpark DataFrame? What are the different types of joins available in PySpark?

In PySpark, joins merge or join two DataFrames together. It facilitates us to link two or multiple DataFrames together.

INNER Join, LEFT OUTER Join, RIGHT OUTER Join, LEFT ANTI Join, LEFT SEMI Join, CROSS Join, and SELF Join are among the SQL join types PySpark supports. Following is the syntax of PySpark Join.

**Syntax:**

1. join(self, other, on=None, how=None)

**Parameter Explanation:**

The join() procedure accepts the following parameters and returns a DataFrame:

* **"other":** It specifies the join's right side.
* **"on":** It specifies the join column's name.
* **"how":** It is used to specify an option. Options are inner, cross, outer, full, full outer, left, left outer, right, right outer, left semi, and left anti. The default is inner.

**Types of Join in PySpark DataFrame**

|  |  |
| --- | --- |
| **Join String** | **Equivalent SQL Join** |
| inner | INNER JOIN |
| outer, full, fullouter, full\_outer | FULL OUTER JOIN |
| left, leftouter, left\_outer | LEFT JOIN |
| right, rightouter, right\_outer | RIGHT JOIN |
| cross |  |
| anti, leftanti, left\_anti |  |
| semi, leftsemi, left\_semi |  |

22) What is Parquet file in PySpark?

In PySpark, the Parquet file is a column-type format supported by several data processing systems. By using the Parquet file, Spark SQL can perform both read and write operations.

The Parquet file contains a column type format storage which provides the following advantages:

* It is small and consumes less space.
* It facilitates us to fetch specific columns for access.
* It follows type-specific encoding.
* It offers better-summarized data.
* It contains very limited I/O operations.

23) What do you understand by a cluster manager? What are the different cluster manager types supported by PySpark?

In PySpark, a cluster manager is a cluster mode platform that facilitates Spark to run by providing all resources to worker nodes according to their requirements.

A Spark cluster manager ecosystem contains a master node and multiple worker nodes. The master nodes provide the worker nodes with the resources like memory, processor allocation, etc., according to the nodes' requirements with the help of the cluster manager.

**PySpark supports the following cluster manager types:**

* **Standalone:** This is a simple cluster manager that comes with Spark.
* **Apache Mesos:** This cluster manager is used to run Hadoop MapReduce and PySpark apps.
* **Hadoop YARN:** This cluster manager is used in Hadoop2.
* **Kubernetes:** This cluster manager is an open-source cluster manager that helps automate deployment, scaling, and automatic management of containerized apps.
* **local:** This cluster manager is a mode for running Spark applications on laptops/desktops.

24) Why is PySpark faster than pandas?

PySpark is faster than pandas because it supports the parallel execution of statements in a distributed environment. For example, PySpark can be executed on different cores and machines, unavailable in Pandas. This is the main reason why PySpark is faster than pandas.

25) What is the difference between get(filename) and getrootdirectory()?

The main difference between get(filename) and getrootdirectory() is that the get(filename) is used to achieve the correct path of the file that is added through SparkContext.addFile(). On the other hand, the getrootdirectory() is used to get the root directory containing the file added through SparkContext.addFile().

26) What do you understand by SparkSession in Pyspark?

In PySpark, SparkSession is the entry point to the application. In the first version of PySpark, SparkContext was used as the entry point. SparkSession is the replacement of SparkContext since PySpark version 2.0. After the PySpark version 2.0, SparkSession acts as a starting point to access all of the PySpark functionalities related to RDDs, DataFrame, Datasets, etc. It is also a Unified API used to replace the SQLContext, StreamingContext, HiveContext, and all other contexts in Pyspark.

The SparkSession internally creates SparkContext and SparkConfig according to the details provided in SparkSession. You can create SparkSession by using builder patterns.

27) What are the key advantages of PySpark RDD?

Following is the list of key advantages of PySpark RDD:

**Immutability:** The PySpark RDDs are immutable. If you create them once, you cannot modify them later. You have to create a new RDD whenever you try to apply any transformation operations on the RDDs.

**Fault Tolerance:** The PySpark RDD provides fault tolerance features. Whenever an operation fails, the data gets automatically reloaded from other available partitions. This provides a seamless experience of execution of the PySpark applications.

**Partitioning:** When we create an RDD from any data, the elements in the RDD are partitioned to the cores available by default.

**Lazy Evolution:** PySpark RDD follows the lazy evolution process. In PySpark RDD, the transformation operations are not performed as soon as they are encountered. The operations would be stored in the DAG and are evaluated once it finds the first RDD action.

**In-Memory Processing:** The PySpark RDD is used to help in loading data from the disk to the memory. You can persist RDDs in the memory for reusing the computations.

28) Explain the common workflow of a spark program.

The common workflow of a spark program can be described in the following steps:

* In the first step, we create the input RDDs depending on the external data. Data can be obtained from different data sources.
* After creating the PySpark RDDs, we run the RDD transformation operations such as filter() or map() to create new RDDs depending on the business logic.
* If we require any intermediate RDDs to reuse for later purposes, we can persist those RDDs.
* Finally, if any action operations like first(), count(), etc., are present, Spark launches it to initiate parallel computation.

29) How can you implement machine learning in Spark?

We can implement machine learning in Spark by using MLlib. Spark provides a scalable machine learning record called MLlib. It is mainly used to create machine learning scalable and straightforward with ordinary learning algorithms and use cases like clustering, weakening filtering, dimensional lessening, etc.

30) What do you understand by custom profilers in PySpark?

PySpark supports custom profilers. The custom profilers are used for building predictive models. Profilers are also used for data review to ensure that it is valid, and we can use it in consumption. When we require a custom profiler, it has to define some of the following methods:

* **stats:** This is used to return collected stats of profiling.
* **profile:** This is used to produce a system profile of some sort.
* **dump:** This is used to dump the profiles to a specified path.
* **dump(id, path):** This is used to dump a specific RDD id to the path given.
* **add:** This is used for adding profile to existing accumulated profile. The profile class has to be selected at the time of SparkContext creation.

31) What do you understand by Spark driver?

The Spark driver is a plan that runs on the master node of a machine. It is mainly used to state actions and alterations on data RDDs.

32) What is PySpark SparkJobinfo?

The PySpark SparkJobinfo is used to get information about the SparkJobs that are in execution.

Following is the code for using the SparkJobInfo:

1. class SparkJobInfo(namedtuple("SparkJobInfo", "jobId stageIds status ")):

33) What are the main functions of Spark core?

The main task of Spark Core is to implement several vital functions such as memory management, fault-tolerance, monitoring jobs, job setting up, and communication with storage systems. It also contains additional libraries, built atop the middle that is used to diverse workloads for streaming, machine learning, and SQL.

**The Spark Core is mainly used for the following tasks:**

* Fault tolerance and recovery.
* To interact with storage systems.
* Memory management.
* Scheduling and monitoring jobs on a cluster.

34) What do you understand by PySpark SparkStageinfo?

The PySpark SparkStageInfo is used to get information about the SparkStages available at that time. Following is the code used for SparkStageInfo:

1. class SparkStageInfo(namedtuple("SparkStageInfo", "stageId currentAttemptId name numTasks unumActiveTasks" "numCompletedTasks numFailedTasks" )):

35) What is the use of Spark execution engine?

The Apache Spark execution engine is a chart execution engine that facilitates users to examine massive data sets with a high presentation. You need to detain Spark in the memory to pick up performance radically if you want data to be manipulated with manifold stages of processing.

36) What is the use of Akka in PySpark?

Akka is used in PySpark for scheduling. When a worker requests a task to the master after registering, the master assigns a task to him. In this case, Akka sends and receives messages between the workers and masters.

37) What do you understand by startsWith() and endsWith() methods in PySpark?

The startsWith() and endsWith() methods in PySpark belong to the Column class and are used to search DataFrame rows by checking if the column value starts with some value or ends with some value. Both are used for filtering data in applications.

* **startsWith() method:** This method is used to return a Boolean value. It shows TRUE when the column's value starts with the specified string and FALSE when the match is not satisfied in that column value.
* **endsWith() method:** This method is used to return a Boolean value. It shows TRUE when the column's value ends with the specified string and FALSE when the match is not satisfied in that column value. Both methods are case-sensitive.

38) What do you understand by RDD Lineage?

The RDD lineage is a procedure that is used to reconstruct the lost data partitions. The Spark does not hold up data replication in the memory. If any data is lost, we have to rebuild it using RDD lineage. This is the best use case as RDD always remembers how to construct from other datasets.

39) Can we create PySpark DataFrame from external data sources?

Yes, we can create PySpark DataFrame from external data sources. The real-time applications use external file systems like local, HDFS, HBase, MySQL table, S3 Azure, etc. The following example shows how to create DataFrame by reading data from a csv file present in the local system:

1. df = spark.read.csv("/path/to/file.csv")

PySpark supports csv, text, avro, parquet, tsv and many other file extensions.

40) What are the main attributes used in SparkConf?

Following is the list of main attributes used in SparkConf:

* **set(key, value):** This attribute is used for setting the configuration property.
* **setSparkHome(value):** This attribute enables the setting Spark installation path on worker nodes.
* **setAppName(value):** This attribute is used for setting the application name.
* **setMaster(value):** This attribute is used to set the master URL.
* **get(key, defaultValue=None):** This attribute supports getting a configuration value of a key.

li>

41) How can you associate Spark with Apache Mesos?

We can use the following steps to associate Spark with Mesos:

* First, configure the sparkle driver program to associate with Mesos.
* The Spark paired bundle must be in the area open by Mesos.
* After that, install Apache Spark in a similar area as Apache Mesos and design the property "spark.mesos.executor.home" to point to the area where it is introduced.

42) What are the main file systems supported by Spark?

Spark supports the following three file systems:

* Local File system.
* Hadoop Distributed File System (HDFS).
* Amazon S3

43) How can we trigger automatic cleanups in Spark to handle accumulated metadata?

We can trigger the automatic cleanups in Spark by setting the parameter ' Spark.cleaner.ttl' or separating the long-running jobs into dissimilar batches and writing the mediator results to the disk.

44) How can you limit information moves when working with Spark?

We can limit the information moves when working with Spark by using the following manners:

* Communicate
* Accumulator factors

45) How is Spark SQL different from HQL and SQL?

Hive is used in HQL (Hive Query Language), and Spark SQL is used in Structured Query language for processing and querying data. We can easily join SQL table and HQL table to Spark SQL. Flash SQL is used as a unique segment on the Spark Core motor that supports SQL and Hive Query Language without changing any sentence structure.

46) What is DStream in PySpark?

In PySpark, DStream stands for Discretized Stream. It is a group of information or gathering of RDDs separated into little clusters. It is also known as Apache Spark Discretized Stream and is used as a gathering of RDDs in the grouping. DStreams are based on Spark RDDs and are used to enable Streaming to flawlessly coordinate with some other Apache Spark segments like Spark MLlib and Spark SQL.

### ****1. How does Azure Synapse differ from Azure Databricks?****

Azure Synapse is a data integration service with some amazing transformation capabilities while Azure Databricks is data analytics focussed platform build on top of Spark. Azure Synapse integrates [big data](https://www.projectpro.io/article/big-data-interview-questions-/773) analytics and enterprise [data warehouse](https://www.projectpro.io/article/data-warehouse-project-ideas-for-practice/572) into a single platform. While Databricks allows customers to develop complex machine learning algorithms and perform big data analytics. However, both both Synapse Analytics and Azure Databricks can be used together when building a [data pipeline](https://www.projectpro.io/article/data-pipeline-definition-architecture-examples/528).

### ****2. Define the linked service in Azure synapse Analytics.****

The external sources outside the Azure Synapse Analytics workspace are connected via Linked Services. A linked service is required in a data pipeline to read or write data to source or destination. For instance, you can use it to establish a connection to the Azure Data Lake storage account to execute [SQL](https://www.projectpro.io/article/sql-database-projects-for-data-analysis-to-practice/565) queries on the files. Suppose you want to pull data from an on-premises server into the cloud or connect to the Azure data lake storage account to perform SQL queries on the files. In these cases, you must build a linked service by providing data such as a username, password, and server address to establish a connection.

### ****3. What do you understand by the default SQL pool in Azure Synapse Analytics?****

Anytime you build an Azure synapse analytics workspace, a serverless SQL pool is already set up. It is referred to as the built-in serverless SQL pool and is a default SQL pool that is accessible for use. The default SQL pool in Azure Synapse Analytics helps with query optimization, data distribution, and data compression.

### ****4. What does the Azure Synapse Analytics OPENROWSET function do?****

The OPENROWSET function lets data engineers read data from diverse data sources including flat files, RDBM’s, and other OLE DB sources. In [Azure](https://www.projectpro.io/article/azure-data-engineering-tools/669) Synapse, one uses the OPENROWSET function to read the file as a table. For instance, you want to perform the queries on a file saved in the ADLS account. Using the rowset function, which converts each row of a file into a row of a table, you can read this file as a table.

### ****5. What makes Azure Synapse Analytics different from Azure Blob storage?****

Large volumes of unstructured data, such as video, audio, and images, can be stored in the cloud using Azure Blob storage. You can gather, store, and analyze huge amounts of data with the help of Azure Synapse Analytics, a cloud-based data warehouse service.In a nutshell, Azure Blob Storage majorly focusses on storing and retrieving data while Azure Synapse Analytics focusses on preparing, managing, and analyzing data.

## **Scenario-based Azure Synapse Analytics Interview Questions**

### ****6. Assume you are the lead data engineer at your company, and a few spark notebooks have been run in the past few hours. You want to examine each notebook execution. Where will you find the past hour few hours' histories and retrieve the event log for them in Azure Synapse Analytics?****

You must access the monitor tab in Azure Synapse Analytics. You can find the activities area in the monitor tab on the left. The Apache Spark application has a separate tab below it. The Spark history server will appear on the following page when you click on it. Simply click on that to view the whole history and to download the event log for a particular application run.

### ****7. Suppose you are a data engineer for the XYZ company. The company wants to transfer its data from an on-site server to the Azure cloud. How will you achieve this using Azure synapse Analytics?****

You must develop an integration runtime to transfer data from an on-premises server to a cloud server. This integration runtime will act as the self-hosted IR since the on-premise servers cannot be connected using the auto-resolve integration runtime, After creating the self-hosted IR, you can use the copy function to build a pipeline that will transfer data from the on-premise server to the cloud server.

### ****8. Suppose you are the lead data engineer at XYZ company, which is transitioning from on-premise to cloud. On one of their on-site servers, your organization holds some mission-critical data. As a result, you must ensure that the team's self-hosted IR is only accessible to specified team members. What is the best way to meet this requirement using Azure Synapse Analytics?****

Open the Azure synapse analytics workspace studio. Navigate to the access control section under the monitor tab. You can grant access at the workspace item level there. Choose the IR item and the IR's name, role, and member data.

### ****9. Assume that a different application will access the data created by your stored procedure in the SQL pool database. Daily updates of this data are necessary. What approach would you apply to address this issue?****

To begin, open the Azure Synapse Analytics workspace and create the pipeline in the integration tab of the resource. The activity for the SQL pool stored procedure must be added to this pipeline. Choose the specified stored procedure from the SQL pool stored procedure activity. Schedule this pipeline by including the trigger that will run it daily.

### ****10. Let's say you want to know how many SQL requests your team has already executed in the SQL pool. Where can you see a list of every historical query run in the Azure Synapse Analytics workspace?****

Any time you perform a query in Azure Synapse Analytics, a history log is generated. You must select the manage tab to see the SQL request. You can sort by time to see the executions and their respective results.

## **Azure Synapse Analytics Interview Questions for Experienced**

### ****11. What does the Azure Synapse Analytics control node do?****

The primary component of a Synapse SQL architecture is the control node. A distributed query engine is executed on the Control node to optimize and coordinate parallel queries. The Control node converts a T-SQL query into parallel queries executed against each distribution when submitted to a specific SQL pool. The DQP engine, used in serverless SQL pools, runs on the Control node and divides user queries into smaller ones that Compute nodes will process to optimize and coordinate distributed execution.

### ****12. What is the method to create a pipeline in the Azure synapse analytics?****

You can create a pipeline in Azure Synapse with the following steps:

* Open the Integrate hub in Synapse Studio.
* Click on 'Pipeline' to build a new pipeline. Open the Pipeline designer by clicking on the newly created pipeline object.
* You can start dragging the activities from the ‘Activities’ panel on the left into the blank space in the middle.
* You must publish your pipeline once it has been created. After that, you can test it to ensure there are no errors and verify that it works.

### ****13. In Azure Synapse Analytics, how much data can you keep in a single column simultaneously?****

The maximum size of a single column in Azure Synapse Analytics depends on the storage technology used and also on the datatype of the column. You can store up to 1 billion 2-byte Unicode characters using nvarchar [(n | max)].

### ****14. Briefly discuss the method for setting up a spark job in Azure Synapse Analytics.****

The Apache Spark job must first be created under the Develop tab. You must now go to the integrate tab and build the pipeline there. Select the spark job activity and include it in the pipeline. Configure the spark job activity by choosing the exact spark job you intend to schedule. Lastly, you must schedule this pipeline, and your Spark task will be scheduled automatically. You can summarize the answer in following steps -

i) Create a Spark Pool

ii) Prepare the Data

iii) Write the Spark Job

iv) Submit the Spark Job

v) Monitor the Spark Job using Synapse Analytics Studio

vi) Schedule the Job

### ****15. Mention some of the data security features offered by Synapse Analytics.****

Synapse Analytics provides certain data security features for dedicated SQL pools, including Data Discovery & Classification, Dynamic Data Masking, Vulnerability Assessment, Advanced Threat Protection, and Transparent Data Encryption.Synapse Analytics supports encryption of data both in transit and at rest. Data can be encrypted using Azure Key Vault, which provides a secure way to store and manage encryption keys.

### ****. Why do we need Azure Data Factory?****

* The amount of data generated these days is huge, and this data comes from different sources. When we move this particular data to the [cloud](https://intellipaat.com/blog/what-is-cloud-computing/), a few things need to be taken care of.
* Data can be in any form, as it comes from different sources. These sources will transfer or channel the data in different ways. They will be in different formats. When we bring this data to the cloud or particular storage, we need to make sure it is well managed, i.e., you need to transform the data and delete unnecessary parts. As far as moving the data is concerned, we need to make sure that data is picked from different sources, brought to one common place, and stored. If required, we should transform it into something more meaningful.
* This can be done by a traditional data warehouse, but there are certain disadvantages. Sometimes we are forced to go ahead and have custom applications that deal with all these processes individually, which is time-consuming, and integrating all these sources is a huge pain. We need to figure out a way to automate this process or create proper workflows.
* Data Factory helps to orchestrate this complete process in a more manageable or organizable manner.

***.***

### ****2. What is Azure Data Factory?****

It is a cloud-based integration service that allows the creation of data-driven workflows in the cloud for orchestrating and automating data movement and transformation.

* Using Azure Data Factory, you can create and schedule data-driven workflows (called pipelines) that ingest data from disparate data stores.
* It can process and transform data using computer services such as HDInsight, Hadoop, Spark, Azure Data Lake Analytics, and [Azure Machine Learning](https://intellipaat.com/blog/tutorial/microsoft-azure-tutorial/azure-machine-learning-ml-tutorial/).

### ****3. What is the integration runtime?****

* The integration runtime is the compute infrastructure Azure Data Factory uses to provide the following data integration capabilities across various network environments.
* Three Types of Integration Runtimes:
  + Azure Integration Runtime: Azure integration runtime (IR) can copy data between cloud data stores and dispatch the activity to a variety of computing services, such as [Azure HDInsight](https://intellipaat.com/blog/what-is-azure-hd-insight/) or SQL Server, where the transformation takes place.
  + Self-Hosted Integration Runtime: A self-hosted integration runtime is software with essentially the same code as Azure integration runtime. But you install it on an on-premise machine or a virtual machine in a virtual network. A self-hosted IR can run copy activities between a public cloud data store and a data store on a private network. It can also dispatch transformation activities against compute resources on a private network. We use self-hosted IR because the Data Factory will not be able to directly access primitive data sources because they sit behind a firewall. It is sometimes possible to establish a direct connection between Azure and on-premises data sources by configuring the [Azure Firewall](https://intellipaat.com/blog/azure-firewall/) in a specific way. If we do that, we don’t need to use a self-hosted IR.
  + Azure-SSIS Integration Runtime: With SSIS integration runtime, you can natively execute SSIS packages in a managed environment. So when we lift and shift the SSIS packages to the Data Factory, we use Azure SSIS IR.

### ****4. What is the limit on the number of integration runtimes?****

There is no hard limit on the number of integration runtime instances you can have in a data factory. There is, however, a limit on the number of VM cores that the integration runtime can use per subscription for SSIS package execution.

Bottom of Form

### ****5. What is the difference between Azure Data Lake and Azure Data Warehouse?****

The [data](https://intellipaat.com/blog/tutorial/data-warehouse-tutorial/data-warehouse-overview/) warehouse is a traditional way of storing data that is still widely used. The data lake is complementary to a data warehouse, i.e., if you have your data in a data lake that can be stored in the data warehouse, you have to follow specific rules.

|  |  |
| --- | --- |
| **DATA LAKE** | **DATA WAREHOUSE** |
| Complementary to the data warehouse | Maybe sourced to the data lake |
| Data is either detailed or raw. It can be in any particular form. You need to take the data and put it in your data lake. | Data is filtered, summarized, and refined. |
| Schema on read (not structured, you can define your schema in n number of ways) | Schema on write (data is written in structured form or a particular schema) |
| One language to process data of any format(USQL) | It uses [SQL](https://intellipaat.com/blog/tutorial/sql-tutorial/introduction-to-sql/). |

**Preparing for the Azure Certification exam? Join our**[**Azure Training in Bangalore**](https://intellipaat.com/microsoft-azure-training-bangalore/)**!**

## Intermediate Interview Questions

### ****6. What is blob storage in Azure?****

[Azure Blob Storage](https://azure.microsoft.com/en-us/services/storage/blobs/) is a service for storing large amounts of unstructured object data, such as text or binary data. You can use Blob Storage to expose data publicly to the world or to store application data privately. Common uses of Blob Storage are as follows:

* Serving images or documents directly to a browser
* Storing files for distributed access
* Streaming video and audio
* Storing data for backup and restore disaster recovery, and archiving
* Storing data for analysis by an on-premises or Azure-hosted service

### ****7. What is the difference between Azure Data Lake store and Blob storage?****

|  |  |  |
| --- | --- | --- |
|  | **Azure Data Lake Storage Gen1** | **Azure Blob Storage** |
| **Purpose** | Optimized storage for big data analytics workloads | General-purpose object store for a wide variety of storage scenarios, including big data analytics |
| **Structure** | Hierarchical file system | Object store with a flat namespace |
| **Key Concepts** | Data Lake Storage Gen1 account contains folders, which in turn contain data stored as files | Storage account has containers, which in turn has data in the form of blobs |
| **Use Cases** | Batch, interactive, streaming analytics, and machine learning data such as log files, IoT data, clickstreams, and large datasets | Any type of text or binary data, such as application back end, backup data, media storage for streaming, and general-purpose data. Additionally, full support for analytics workloads: batch, interactive, streaming analytics, and machine learning data such as log files, IoT data, clickstreams, and large datasets |
| **Server-Side API** | [WebHDFS-compatible REST API](https://docs.microsoft.com/en-us/rest/api/datalakestore/) | [Azure Blob Storage REST API](https://docs.microsoft.com/en-us/rest/api/storageservices/Blob-Service-REST-API) |
| **Data Operations – Authentication** | Based on [Azure Active Directory Identities](https://docs.microsoft.com/en-us/azure/active-directory/develop/authentication-scenarios) | Based on shared secrets – [Account Access Keys](https://docs.microsoft.com/en-us/azure/storage/common/storage-account-manage#access-keys) and [Shared Access Signature Keys](https://docs.microsoft.com/en-us/azure/storage/common/storage-dotnet-shared-access-signature-part-1). |

***To learn more about big data, check out this***[***Big Data Course***](https://intellipaat.com/big-data-hadoop-training/)***offered by Intellipaat.***

### ****8. What are the steps for creating ETL process in Azure Data Factory?****

While we are trying to extract some data from the Azure SQL Server database, if something has to be processed, it will be processed and stored in the Data Lake Storage.

**Steps for Creating ETL**

* Create a linked service for the source data store, which is SQL Server Database
* Assume that we have a cars dataset
* Create a linked service for the destination data store, which is [Azure Data Lake](https://intellipaat.com/blog/what-is-azure-data-lake/) Storage (ADLS)
* Create a dataset for data saving
* Create the pipeline and add copy activity
* Schedule the pipeline by adding a trigger

### ****9. What is the difference between HDInsight and Azure Data Lake Analytics?****

|  |  |
| --- | --- |
| **HDInsight** | **Azure Data Lake Analytics** |
| If we want to process a data set, first of all, we have to configure the cluster with predefined nodes, and then we use a language like Pig or Hive for processing the data. | It is all about passing queries written for processing data. Azure Data Lake Analytics will create the necessary compute nodes per our instructions on demand and process the data set. |
| Since we configure the cluster with HDInsight, we can create it as we want and control it as we want. All [Hadoop](https://intellipaat.com/blog/what-is-hadoop/) subprojects, such as Spark and [Kafka](https://intellipaat.com/blog/what-is-apache-kafka/), can be used without limitations. | With Azure Data Lake Analytics, it does not give much flexibility in terms of the provision in the cluster, but [Microsoft Azure](https://intellipaat.com/blog/what-is-microsoft-azure/) takes care of it. We don’t need to worry about cluster creation. The assignment of nodes will be done based on the instructions we pass. In addition, we can make use of U-SQL taking advantage of .Net for processing data. |

### ****10. What are the top-level concepts of Azure Data Factory?****

* **Pipeline:** It acts as a carrier in which various processes take place. An individual process is an activity.
* **Activities:** Activities represent the processing steps in a pipeline. A pipeline can have one or multiple activities. It can be anything, i.e., a process like querying a data set or moving the dataset from one source to another.
* **Datasets:** In simple words, it is a data structure that holds our data.
* **Linked Services:** These store information that is very important when connecting to an external source.

For example, consider an SQL Server. You need a connection string that you can connect to an external device. You need to mention the source and destination of your data.

## Advanced Interview Questions

### ****11. How can I schedule a pipeline?****

* You can use the scheduler trigger or time window trigger to schedule a pipeline.
* The trigger uses a wall-clock calendar schedule, which can schedule pipelines periodically or in calendar-based recurrent patterns (for example, on Mondays at 6:00 PM and Thursdays at 9:00 PM).

### ****12. Can I pass parameters to a pipeline run?****

* Yes, parameters are a first-class, top-level concept in Data Factory.
* You can define parameters at the pipeline level and pass arguments as you execute the pipeline run-on-demand or by using a trigger.

**Are you looking to learn about Azure? Check out our blog on**[**Azure Tutorial**](https://intellipaat.com/blog/tutorial/microsoft-azure-tutorial/)**!**

### ****13. Can I define default values for the pipeline parameters?****

You can define default values for the parameters in the pipelines.

### ****14. Can an activity in a pipeline consume arguments that are passed to a pipeline run?****

In a pipeline, an activity can indeed consume arguments that are passed to a pipeline run. Arguments serve as input values that can be provided when triggering or scheduling a pipeline run. These arguments can be used by activities within the pipeline to customize their behavior or perform specific tasks based on the provided values. This flexibility allows for dynamic and parameterized execution of pipeline activities, enhancing the versatility and adaptability of the pipeline workflow.

Each activity within the pipeline can consume the parameter value that’s passed to the pipeline and run with the **@parameter construct.**

### ****15. Can an activity’s output property be consumed in another activity?****

An activity output can be consumed in a subsequent activity with the @activity construct.

### ****16. How do I handle null values in an activity output?****

You can use the **@coalesce construct** in the expressions to handle the null values.

***Check out Intellipaat’s***[***Azure Training***](https://intellipaat.com/microsoft-azure-training/)***and get a head start in your career now!***

### ****17. Which Data Factory version do I use to create data flows?****

Use the Data Factory version 2 to create data flows.

### ****18. What has changed from private preview to limited public preview in regard to data flows?****

* You will no longer have to bring your own Azure Databricks clusters.
* Data Factory will manage cluster creation and teardown.
* Blob datasets and Azure Data Lake Storage Gen2 datasets are separated into delimited text and Apache Parquet datasets.
* You can still use Data Lake Storage Gen2 and Blob Storage to store those files. Use the appropriate linked service for those storage engines.

### ****19. How do I access data using the other 80 dataset types in Data Factory?****

* The mapping data flow feature currently allows Azure SQL Database, Azure SQL Data Warehouse, delimited text files from Azure Blob Storage or Azure Data Lake Storage Gen2, and Parquet files from Blob Storage or Data Lake Storage Gen2 natively for source and sink.
* Use the copy activity to stage data from any of the other connectors, and then execute a Data Flow activity to transform the data after it’s been staged. For example, your pipeline will first copy into Blob Storage, and then a Data Flow activity will use a dataset in the source to transform that data.

***Learn about various certifications in Azure in our in-depth blog on***[***Microsoft Azure Certification***](https://intellipaat.com/blog/microsoft-azure-certification/)***.***

### ****20. Explain the two levels of security in ADLS Gen2.****

The two levels of security applicable to ADLS Gen2 were also in effect for ADLS Gen1. Even though this is not new, it is worth calling out the two levels of security because it’s a fundamental piece to getting started with the data lake, and it is confusing for many people to start.

* Role-Based Access Control (RBAC). RBAC includes built-in Azure roles such as reader, contributor, owner, or custom roles. Typically, RBAC is assigned for two reasons. One is to specify who can manage the service itself (i.e., update settings and properties for the storage account). Another reason is to permit the use of built-in data explorer tools, which require reader permissions.
* Access Control Lists (ACLs). Access control lists specify exactly which data objects a user may read, write, or execute (execute is required to browse the directory structure). ACLs are POSIX-compliant, thus familiar to those with a Unix or Linux background.