Free Questions for Databricks-Certified-Associate-Developer-for-Apache-Spark-3.0 by dumpssheet

Shared by Flynn on 15-04-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

Which of the following describes Spark actions?

Options:

A- Writing data to disk is the primary purpose of actions.

B- Actions are Spark's way of exchanging data between executors.

C- The driver receives data upon request by actions.

D- Stage boundaries are commonly established by actions.

E- Actions are Spark's way of modifying RDDs.

Answer:

C

Explanation:

The driver receives data upon request by actions.

Correct! Actions trigger the distributed execution of tasks on executors which, upon task completion, transfer result data back to the driver.

Actions are Spark's way of exchanging data between executors.

No. In Spark, data is exchanged between executors via shuffles.

Writing data to disk is the primary purpose of actions.

No. The primary purpose of actions is to access data that is stored in Spark's RDDs and return the data, often in aggregated form, back to the driver.

Actions are Spark's way of modifying RDDs.

Incorrect. Firstly, RDDs are immutable -- they cannot be modified. Secondly, Spark generates new RDDs via transformations and not actions.

Stage boundaries are commonly established by actions.

Wrong. A stage boundary is commonly established by a shuffle, for example caused by a wide transformation.

Question 2

Question Type: MultipleChoice

Which of the following are valid execution modes?

Options:

A- Kubernetes, Local, Client

B- Client, Cluster, Local

C- Server, Standalone, Client

D- Cluster, Server, Local

E- Standalone, Client, Cluster

Answer:

B

Explanation:

This is a tricky Question: to get right, since it is easy to confuse execution modes and deployment modes. Even in literature, both terms are sometimes used interchangeably.

There are only 3 valid execution modes in Spark: Client, cluster, and local execution modes. Execution modes do not refer to specific frameworks, but to where infrastructure is located with respect

to each other.

In client mode, the driver sits on a machine outside the cluster. In cluster mode, the driver sits on a machine inside the cluster. Finally, in local mode, all Spark infrastructure is started in a single JVM

(Java Virtual Machine) in a single computer which then also includes the driver.

Deployment modes often refer to ways that Spark can be deployed in cluster mode and how it uses specific frameworks outside Spark. Valid deployment modes are standalone, Apache YARN,

Apache Mesos and Kubernetes.

Client, Cluster, Local

Correct, all of these are the valid execution modes in Spark.

Standalone, Client, Cluster

No, standalone is not a valid execution mode. It is a valid deployment mode, though.

Kubernetes, Local, Client

No, Kubernetes is a deployment mode, but not an execution mode.

Cluster, Server, Local

No, Server is not an execution mode.

Server, Standalone, Client

No, standalone and server are not execution modes.

More info: Apache Spark Internals - Learning Journal

Question 3

Question Type: MultipleChoice

Which of the following is a characteristic of the cluster manager?

Options:

A- Each cluster manager works on a single partition of data.

B- The cluster manager receives input from the driver through the SparkContext.

C- The cluster manager does not exist in standalone mode.

D- The cluster manager transforms jobs into DAGs.

E- In client mode, the cluster manager runs on the edge node.

Answer:

B

Explanation:

The cluster manager receives input from the driver through the SparkContext.

Correct. In order for the driver to contact the cluster manager, the driver launches a SparkContext. The driver then asks the cluster manager for resources to launch executors.

In client mode, the cluster manager runs on the edge node.

No. In client mode, the cluster manager is independent of the edge node and runs in the cluster.

The cluster manager does not exist in standalone mode.

Wrong, the cluster manager exists even in standalone mode. Remember, standalone mode is an easy means to deploy Spark across a whole cluster, with some limitations. For example, in

standalone mode, no other frameworks can run in parallel with Spark. The cluster manager is part of Spark in standalone deployments however and helps launch and maintain resources across the

cluster.

The cluster manager transforms jobs into DAGs.

No, transforming jobs into DAGs is the task of the Spark driver.

Each cluster manager works on a single partition of data.

No. Cluster managers do not work on partitions directly. Their job is to coordinate cluster resources so that they can be requested by and allocated to Spark drivers.

More info: Introduction to Core Spark Concepts * BigData

Question 4

Question Type: MultipleChoice

Which of the following statements about the differences between actions and transformations is correct?

Options:

A- Actions are evaluated lazily, while transformations are not evaluated lazily.

B- Actions generate RDDs, while transformations do not.

C- Actions do not send results to the driver, while transformations do.

D- Actions can be queued for delayed execution, while transformations can only be processed immediately.

E- Actions can trigger Adaptive Query Execution, while transformation cannot.

Answer:

E

Explanation:

Actions can trigger Adaptive Query Execution, while transformation cannot.

Correct. Adaptive Query Execution optimizes queries at runtime. Since transformations are evaluated lazily, Spark does not have any runtime information to optimize the query until an action is

called. If Adaptive Query Execution is enabled, Spark will then try to optimize the query based on the feedback it gathers while it is evaluating the query.

Actions can be queued for delayed execution, while transformations can only be processed immediately.

No, there is no such concept as 'delayed execution' in Spark. Actions cannot be evaluated lazily, meaning that they are executed immediately.

Actions are evaluated lazily, while transformations are not evaluated lazily.

Incorrect, it is the other way around: Transformations are evaluated lazily and actions trigger their evaluation.

Actions generate RDDs, while transformations do not.

No. Transformations change the data and, since RDDs are immutable, generate new RDDs along the way. Actions produce outputs in Python and data types (integers, lists, text files,...) based on

the RDDs, but they do not generate them.

Here is a great tip on how to differentiate actions from transformations: If an operation returns a DataFrame, Dataset, or an RDD, it is a transformation. Otherwise, it is an action.

Actions do not send results to the driver, while transformations do.

No. Actions send results to the driver. Think about running DataFrame.count(). The result of this command will return a number to the driver. Transformations, however, do not send results back to

the driver. They produce RDDs that remain on the worker nodes.

More info: What is the difference between a transformation and an action in Apache Spark? | Bartosz Mikulski, How to Speed up SQL Queries with Adaptive Query Execution

Question 5

Question Type: MultipleChoice

Which of the following describes properties of a shuffle?

Options:

A- Operations involving shuffles are never evaluated lazily.

B- Shuffles involve only single partitions.

C- Shuffles belong to a class known as 'full transformations'.

D- A shuffle is one of many actions in Spark.

E- In a shuffle, Spark writes data to disk.

Answer:

E

Explanation:

In a shuffle, Spark writes data to disk.

Correct! Spark's architecture dictates that intermediate results during a shuffle are written to disk.

A shuffle is one of many actions in Spark.

Incorrect. A shuffle is a transformation, but not an action.

Shuffles involve only single partitions.

No, shuffles involve multiple partitions. During a shuffle, Spark generates output partitions from multiple input partitions.

Operations involving shuffles are never evaluated lazily.

Wrong. A shuffle is a costly operation and Spark will evaluate it as lazily as other transformations. This is, until a subsequent action triggers its evaluation.

Shuffles belong to a class known as 'full transformations'.

Not quite. Shuffles belong to a class known as 'wide transformations'. 'Full transformation' is not a relevant term in Spark.

More info: Spark -- The Definitive Guide, Chapter 2 and Spark: disk I/O on stage boundaries explanation - Stack Overflow

Question 6

Question Type: MultipleChoice

Which of the following describes Spark's standalone deployment mode?

Options:

A- Standalone mode uses a single JVM to run Spark driver and executor processes.

B- Standalone mode means that the cluster does not contain the driver.

C- Standalone mode is how Spark runs on YARN and Mesos clusters.

D- Standalone mode uses only a single executor per worker per application.

E- Standalone mode is a viable solution for clusters that run multiple frameworks, not only Spark.

Answer:

D

Explanation:

Standalone mode uses only a single executor per worker per application.

This is correct and a limitation of Spark's standalone mode.

Standalone mode is a viable solution for clusters that run multiple frameworks.

Incorrect. A limitation of standalone mode is that Apache Spark must be the only framework running on the cluster. If you would want to run multiple frameworks on the same cluster in parallel, for

example Apache Spark and Apache Flink, you would consider the YARN deployment mode.

Standalone mode uses a single JVM to run Spark driver and executor processes.

No, this is what local mode does.

Standalone mode is how Spark runs on YARN and Mesos clusters.

No. YARN and Mesos modes are two deployment modes that are different from standalone mode. These modes allow Spark to run alongside other frameworks on a cluster. When Spark is run in

standalone mode, only the Spark framework can run on the cluster.

Standalone mode means that the cluster does not contain the driver.

Incorrect, the cluster does not contain the driver in client mode, but in standalone mode the driver runs on a node in the cluster.

More info: Learning Spark, 2nd Edition, Chapter 1

Question 7

Question Type: MultipleChoice

Which of the following describes a difference between Spark's cluster and client execution modes?

Options:

A- In cluster mode, the cluster manager resides on a worker node, while it resides on an edge node in client mode.

B- In cluster mode, executor processes run on worker nodes, while they run on gateway nodes in client mode.

C- In cluster mode, the driver resides on a worker node, while it resides on an edge node in client mode.

D- In cluster mode, a gateway machine hosts the driver, while it is co-located with the executor in client mode.

E- In cluster mode, the Spark driver is not co-located with the cluster manager, while it is co-located in client mode.

Answer:

C

Explanation:

In cluster mode, the driver resides on a worker node, while it resides on an edge node in client mode.

Correct. The idea of Spark's client mode is that workloads can be executed from an edge node, also known as gateway machine, from outside the cluster. The most common way to execute Spark

however is in cluster mode, where the driver resides on a worker node.

In practice, in client mode, there are tight constraints about the data transfer speed relative to the data transfer speed between worker nodes in the cluster. Also, any job in that is executed in client

mode will fail if the edge node fails. For these reasons, client mode is usually not used in a production environment.

In cluster mode, the cluster manager resides on a worker node, while it resides on an edge node in client execution mode.

No. In both execution modes, the cluster manager may reside on a worker node, but it does not reside on an edge node in client mode.

In cluster mode, executor processes run on worker nodes, while they run on gateway nodes in client mode.

This is incorrect. Only the driver runs on gateway nodes (also known as 'edge nodes') in client mode, but not the executor processes.

In cluster mode, the Spark driver is not co-located with the cluster manager, while it is co-located in client mode.

No, in client mode, the Spark driver is not co-located with the driver. The whole point of client mode is that the driver is outside the cluster and not associated with the resource that manages the

cluster (the machine that runs the cluster manager).

In cluster mode, a gateway machine hosts the driver, while it is co-located with the executor in client mode.

No, it is exactly the opposite: There are no gateway machines in cluster mode, but in client mode, they host the driver.

Question 8

Question Type: **MultipleChoice**

Which of the following describes tasks?

Options:

A- A task is a command sent from the driver to the executors in response to a transformation.

B- Tasks transform jobs into DAGs.

C- A task is a collection of slots.

D- A task is a collection of rows.

E- Tasks get assigned to the executors by the driver.

Answer:

E

Explanation:

Tasks get assigned to the executors by the driver.

Correct! Or, in other words: Executors take the tasks that they were assigned to by the driver, run them over partitions, and report the their outcomes back to the driver.

Tasks transform jobs into DAGs.

No, this statement disrespects the order of elements in the Spark hierarchy. The Spark driver transforms jobs into DAGs. Each job consists of one or more stages. Each stage contains one or more

tasks.

A task is a collection of rows.

Wrong. A partition is a collection of rows. Tasks have little to do with a collection of rows. If anything, a task processes a specific partition.

A task is a command sent from the driver to the executors in response to a transformation.

Incorrect. The Spark driver does not send anything to the executors in response to a transformation, since transformations are evaluated lazily. So, the Spark driver would send tasks to executors

only in response to actions.

A task is a collection of slots.

No. Executors have one or more slots to process tasks and each slot can be assigned a task.

Question 9

Question Type: MultipleChoice

Which of the following statements about stages is correct?

Options:

A- Different stages in a job may be executed in parallel.

B- Stages consist of one or more jobs.

C- Stages ephemerally store transactions, before they are committed through actions.

D- Tasks in a stage may be executed by multiple machines at the same time.

E- Stages may contain multiple actions, narrow, and wide transformations.

Answer:

D

Explanation:

Tasks in a stage may be executed by multiple machines at the same time.

This is correct. Within a single stage, tasks do not depend on each other. Executors on multiple machines may execute tasks belonging to the same stage on the respective partitions they are

holding at the same time.

Different stages in a job may be executed in parallel.

No. Different stages in a job depend on each other and cannot be executed in parallel. The nuance is that every task in a stage may be executed in parallel by multiple machines.

For example, if a job consists of Stage A and Stage B, tasks belonging to those stages may not be executed in parallel. However, tasks from Stage A may be executed on multiple machines at the

same time, with each machine running it on a different partition of the same dataset. Then, afterwards, tasks from Stage B may be executed on multiple machines at the same time.

Stages may contain multiple actions, narrow, and wide transformations.

No, stages may not contain multiple wide transformations. Wide transformations mean that shuffling is required. Shuffling typically terminates a stage though, because data needs to be exchanged

across the cluster. This data exchange often causes partitions to change and rearrange, making it impossible to perform tasks in parallel on the same dataset.

Stages ephemerally store transactions, before they are committed through actions.

No, this does not make sense. Stages do not 'store' any data. Transactions are not 'committed' in Spark.

Stages consist of one or more jobs.

No, it is the other way around: Jobs consist of one more stages.

More info: Spark: The Definitive Guide, Chapter 15.