# Your grade: 100%

Your latest: **100%** • Your highest: **100%** • To pass you need at least 70%. We keep your highest score.

Next item

**1.** Which tabs are included within the Apache Spark User Interface? 1 / 1 point

- ○ Jobs, stages, storage, environment, and SQL
- ◉ Jobs, stages, storage, environment, executor, and SQL
- ○ Jobs, stages, storage, executor, and SQL
- ○ Jobs, storage, environment, executor, and SQL

**Correct**
Correct! Jobs, stages, storage, environment, executor, and SQL tabs are available within the Spark Application UI. However, the SQL tab is optional and displayed based on the application.

**2.** What happens during job progression if any tasks within a stage fail after several attempts? 1 / 1 point

- ◉ Apache Spark marks the task, stage, and job as failed and stops the application
- ○ Apache Spark assigns the jobs to other applications
- ○ Apache Spark transfers the jobs to the driver
- ○ Apache Spark sends the jobs to the dependent stage

**Correct**
Correct! If any of the tasks within a stage fail, after several attempts, Apache Spark marks the task, stage, and job as failed and stops the application.

**3.** Syntax, serialization, data validation, and other user errors can occur when running Apache Spark applications. Consider the following numbered list: 1 / 1 point

1. View the driver event log to locate the cause of an application failure.
2. If all attempts to run the task fail, Apache Spark reports an error to the driver, and the application is terminated.
3. If a task fails due to an error, Apache Spark can attempt to rerun the task for a set number of retries.

Select the option that places this list in the order of how Apache Spark handles application errors.

- ◉ 3, 2, 1
- ○ 1, 2, 3
- ○ 2, 1, 3
- ○ 3, 1, 2

**Correct**
Correct! You can use Apache Spark to rerun tasks. If all attempts to run the task fail, Spark reports an error to the driver, and the application is terminated. You can then view the driver event log to locate the cause of an application failure.

**4.** Which of the following is true regarding data persistence? 1 / 1 point

- ○ Leads to out-of-memory errors
- ○ Shares a unified region in the Java Heap Space
- ◉ Stores intermediate calculations for reuse
- ○ Enables an Apache Spark application to run without using all the available cluster memory

**Correct**

Correct! Data persistence, or caching data, in Apache Spark means being able to store intermediate calculations for reuse.

**5.** Which command specifies the number of executor cores for an Apache Spark Standalone cluster per executor process? **1 / 1 point**

- (•) Use the command `--executor-cores` followed by the number of cores.
- ( ) Use the command `--process--executor—cores` followed by the number of cores
- ( ) Use the command `--per--executor—cores` followed by the number of cores.
- ( ) Use the command `--executor-process-cores` followed by the number of cores

**Correct**
Correct! Only Spark Standalone, YARN, and Kubernetes support this argument. The value shows how many cores each executor is using. In YARN and Kubernetes modes, the default value is 1, but in stand-alone mode, it is all of the worker's available cores.

**6.** Which workflow options can you monitor using the Apache Spark application UI? **1 / 1 point**

- ( ) Jobs assigned to other applications.
- (•) Jobs in progress running as tasks in the executors
- ( ) Incomplete Jobs transferring results back to the cluster
- ( ) Jobs created by the SparkContext in the executer program

**Correct**
Correct! The Apache Spark application UI monitors these jobs.

**7.** Which of the following statements is true? **1 / 1 point**

- (•) Workers in the cluster contain a limited number of cores.
- ( ) Workers in the cluster contain unlimited cores.
- ( ) Spark needs permissions to assign CPU cores to driver and executor processes during application processing.
- ( ) If no cores are available to an application, the application automatically processes the tasks

**Correct**
Correct! Workers in the cluster contain a limited number of cores.