

Steps to migrate from IzODA Spark (Spark 2.4) to IBM Z Platform for Apache Spark (Spark 3.x)

1. Read through the entirety of these instructions before starting. Knowing the information needed in later steps can often inform and improve the decisions made in the earlier steps.
2. Obtain all prerequisites for IBM Z Platform for Apache Spark. These will be listed in the Program Directory. Special consideration should be given to:
 - a. Bash 4.3.48 (or higher). Directions for obtaining bash are within the Installation and Customization Guide.
 - b. Java 8 or Java 11. (Java 11 requires HSPK130 APAR, PH54122.)
 - c. Optional: IBM Enterprise Python
3. Obtain and install the IBM Z Platform for Apache Spark product (5698-SPK)
 - a. Notable differences between IBM Open Data Analytics for z/OS Spark and IBM Z Platform for Apache Spark:
 - i. By default, the new Spark product will install to `/usr/lpp/IBM/zspark/` with a resulting `SPARK_HOME` directory of `/usr/lpp/IBM/zspark/spark/spark32x`
 - ii. Sample started task JCL is available within the `SPARK_HOME` directory, in `$SPARK_HOME/samples/zos/jcl`, instead of a separate `SAMPLIB` data set.
 - iii. The product component prefix was "AZK" and is now "AFK", the component ID is HSPK130 (rather than HSPK120).
 - iv. This version uses Log4J V2. The configuration options for Log4J2 are set in `$SPARK_CONF_DIR/log4j2.properties`, which can be copied from `$SPARK_HOME/conf/log4j2.properties.template`
 - b. Product documentation can be found at <https://www.ibm.com/docs/en/zpas/1.1.0>
4. Choose whether to use the existing configuration options and logging locations, or create a new installation with new directories, new port numbers assigned, new Spark job names, new Spark "SPARKID" master user, new RACF resource names. (Note that some RACF updates will be required, regardless.)
 - a. Considerations:
 - i. Upgrading in place is probably easiest, but will require users to update their configuration and applications (recompile) after the cluster master/worker is updated. It will also be more difficult to test without affecting end users. More on this below.
 - ii. If you used the existing naming convention for Spark started tasks (e.g AZKMSTR, etc.), this may become confusing with the new spark product. The Spark z/OS component prefix of "AZK" for IzODA Spark is now "AFK" for IBM Z Platform for Apache Spark.
5. IzODA Spark and the IBM Z Platform for Apache Spark contain an exclusive "spark-configuration-checker" that will display and verify the values and consistency of the Spark configuration. Its use is documented in the Installation & Customization Guide for

the product you are using. Note that it must be used from the userid that is running the spark process(es), and works best from a SSH terminal running the Bash shell. We recommend using it now from the userid that runs the Spark daemons (Spark master, Spark Worker) and recording the results using the second example here.

For example, to see the values used for a given configuration, you would execute `$SPARK_HOME/sbin/spark-configuration-checker.sh -v -c` (Partial output shown):

```
SBJ@PLPSC:~>$SPARK_HOME/sbin/spark-configuration-checker.sh -v -c
===== CHECKS =====
Pass    Obtaining environment variables
Pass    Checking SPARK_HOME
Pass    Obtaining SPARK_CONF_DIR
Pass    Checking if SPARK_CONF_DIR is default
Pass    Checking if SPARK_CONF_DIR is a directory
Pass    Checking if SPARK_CONF_DIR has correct permissions
Pass    Checking that spark-env.sh exists
```

To record the values in a file for later reference, this command will redirect the output to a USS file named:

```
$SPARK_HOME/sbin/spark-configuration-checker.sh -v > cfg.txt
```

Note that the "-c" parameter is left off to ensure no extra color encoding is added to the resulting "cfg.txt" file.

6. If upgrading in place and reusing the directories (conf, logs, work, etc.) and port numbers, continue to 6a. If creating a separate installation and customization, skip to Step 14. (You may want to read the intervening steps as a summary for what will need to be done for the new installation.)
 - a. To-do: started task procedure updates
 - i. Make a backup copy of existing Spark started task procedures:
Required: AZKMSTR, AZKWRKR
Optional: AZKHIST
 - ii. Using TSO OGET or ISPF cut/paste, copy the started task JCL for AFKMSTR, AFKWRKR and (optionally) AFKHIST from ``$SPARK_HOME/samples/zos/jcl`` to your PROCLIB data set.
 1. Example (using ISPF option 6)

```
ISPF Command Shell
ISPF Command ==>
Enter TSO or Workstation commands below:
==> oget
'/usr/lpp/IBM/zspark/spark/spark32x/samples/zos/jcl/afkmstr.jcl'
'sbj.zspark.test.proclib(afkmstr)' text
```

7. The samples are shipped in EBCDIC, so you should not need the CONVERT keyword on the TSO OGET. The TEXT keyword is also unneeded but added here for clarity.
8. Repeat for the other two included sample PROCLIB JCL members.
9. Disable external shuffle server by removing it from COMMNDxx or other automation. (In Apache Spark 3.x, the shuffle server will be part of the Spark Worker in all supported IBM Z Platform for Apache Spark MVS environments.)

10. Update the Spark configuration

- a. Locate the SPARK_CONF_DIR setting for the existing IzODA Spark installation.
- b. Update the "export SPARK_CONF_DIR=" statement in each of the JCL procedures to point to your installation's Spark conf directory.
- c. Update SPARK_HOME environment variable (See 1.a.i above) in the \$SPARK_CONF_DIR/spark-zos-started-tasks.sh file, if necessary.
- d. As noted above, you must use the new `log4j2.properties` file which has a different format than in log4j v1. To do this, copy the template file to your Spark conf directory. For example, if your installation's Spark conf directory is /etc/zspark/spark/conf, you could copy the file using the following USS command:

```
cp $SPARK_HOME/conf/log4j2.properties.template  
/etc/zspark/spark/conf/
```

 - i. If you have made any changes to the Spark "log4j.properties" file, you will need to make those same changes to the new file, adjusting for the syntax changes. See the following reference for details:
Reference: <https://stackoverflow.com/questions/35900555/migrating-from-log4j-to-log4j2-properties-file-configuration> .

11. z/OS updates

- a. Since you are reusing the IzODA Spark configuration, Spark configuration is done. However, TCPIP and RACF updates might be needed.
- b. TCPIP Port adjustments
 - i. Many customers "lock down" their TCPIP ports, only allowing specific jobnames to access specific TCPIP port numbers. If the spark jobnames for the master, worker or history server have changed, updates to the TCPIP PORTS files to include those job names may be necessary
 1. DO NOT use the SHAREPORT keyword on IzODA Spark or IBM Z Platform for Apache Spark port specifications. This is documented in the troubleshooting section of the Installation and Customization Guide.

12. ATTLS and Trusted Partner:

- a. If you are using the same ports for zSpark as you were for IzODA Spark, you will likely be able to use the same ATTLS/TrustedPartner PAGENT definitions.
- b. If you will be using different port numbers, it is recommended that you use the normal configuration steps provided in the Installation and Customization Guide.

13. SAF updates

- a. Use RLIST to find users of the old resource, (AZK.SPARK.MASTER.CONNECT)
RLIST XFACILIT (AZK.SPARK.MASTER.CONNECT) AUTHUSER
- b. Add the new ZSpark SAF resource. For RACF, this would be:
RDEFINE XFACILIT AFK.SPARK.MASTER.CONNECT UACC(NONE)
then permit the users listed above to the new ZSpark SAF resource:
PERMIT AFK.SPARK.MASTER.CONNECT ID(SPARKUSR)
CLASS(XFACILIT) ACC(READ)

- c. If you created new started task procs, you will need to associate those with the SPARKID userid:

```
RDEFINE STARTED AFKMSTR.* STDATA(USER(SPARKID)
GROUP(SYS1))
```

```
RDEFINE STARTED AFKWRKR.* STDATA(USER(SPARKID)
GROUP(SYS1))
```

```
RDEFINE STARTED AFKHIST.* STDATA(USER(SPARKID)
GROUP(SYS1))
```

14. If creating a new Spark installation, use the procedures listed in the Installation and Customization Guide to create a configuration specific to ZSpark.

Note: the steps above will help you locate the changes you made to IzODA environment to properly configure zSpark, as documented in the Installation and Customization Guide.

15. IVT

Regardless of whether you reused your customized IzODA Spark configuration or created a new Spark configuration, you can use the steps documented in [Chapter 5. Verifying the IBM Z Platform for Apache Spark customization](#) of the IBM Z Platform for Apache Spark Installation and Customization Guide (GI13-5809-00)

16. User updates:

- a. Inform the user of the new product and configuration details. They will need to update their environment variables including SPARK_HOME, SPARK_CONF_DIR, SPARK_LOG_DIR, etc.
 - b. In addition, if you are also converting from IzODA MDS to DVM, users will need to procure the appropriate DVM JDBC driver to access DVM. If any virtual tables are renamed, they will need that information to update the parameters being passed to their JDBC driver in their code. See the MDS migration documentation on this server for more information.
- In addition to the installation and customization updates needed to move from IzODA Spark to IBM Z Platform for Apache Spark, the underlying version of Spark is changing from Spark 2.4 to Spark 3.2. Many new features were introduced, including some API changes that require Spark applications to be recompiled. Users should be notified about the version change. The Apache spark migration guide can be found here: <https://spark.apache.org/docs/3.2.0/migration-guide.html>