S3\_JARS = "s3://aml-quantexa-scoring-s3/jars"

S3\_INPUT = "s3://aml-quantexa-scoring-s3/data/quantexa/etl/input/file\_name"

S3\_OUTPUT = "s3://aml-quantexa-scoring-s3/data/quantexa/etl/output/file\_name"

spark\_job = KubernetesPodOperator(

task\_id="spark\_job",

name="spark-submit-job",

namespace="spark-job",

image="891376921217.dkr.ecr.us-east-1.amazonaws.com/spark:3.3.1",

cmds=["/opt/spark/bin/spark-submit"],

arguments=[

"--master", "k8s://https://<api-server>:443",

"--deploy-mode", "cluster",

"--class", "com.quantexa.project.max.transaction.etl.GenerateMetadataRow",

"--conf", "spark.sql.shuffle.partitions=12001",

"--conf", "spark.sql.adaptive.enabled=true",

f"{S3\_JARS}/quantexa-data-source-all-shadow-2.6.9-dependency.jar",

"-c", "./external.conf",

"-r", "etl",

"-l", "qssMetrics",

S3\_INPUT,

S3\_OUTPUT,

],

get\_logs=True,

is\_delete\_operator\_pod=True

)

The JAR really contains the class  
com.quantexa.project.max.transaction.etl.GenerateMetadataRow must exist inside  
quantexa-data-source-all-shadow-2.6.9-dependency.jar and must have a public static void main(String[] args) method.

 No dependency on runQSS.sh

* If runQSS.sh is only a wrapper that calls spark-submit --class … → then yes, removing it and adding --class directly (as in my code) will work.
* If runQSS.sh is doing extra setup (adding configs, classpaths, env vars, etc.), then running the JAR directly may fail unless you replicate those steps in your DAG.

Config file is accessible  
./external.conf must exist inside the working directory of the Spark driver pod. If it’s not packaged in the image, you’ll need to mount it (e.g., from ConfigMap, Secret, or S3).

S3 paths accessible  
Since you’re on EKS with IRSA, make sure your pod’s IAM role has S3 read/write permissions for:

* s3://aml-quantexa-scoring-s3/jars/...
* s3://aml-quantexa-scoring-s3/data/...