

Apache Spark Cheatsheet

Connection

Start Spark shell
spark-shell

Start PySpark shell
pyspark

Start Spark with specific configuration
spark-shell --conf spark.executor.memory=2g

Submit a Spark application
spark-submit --class org.example.MyApp --master yarn app.jar

Start Spark with specified master
spark-shell --master spark://host:port

Monitoring

View Spark UI
<http://localhost:4040> (or cluster-specific URL)

View Spark History Server
<http://spark-history-server:18080>

Monitor active applications
yarn application -list (when using YARN)

View Spark logs
yarn logs -applicationId <app_id> (when using YARN)

Check Spark driver logs
cat /var/log/spark/spark-driver.log (location varies by deployment)

Performance Tuning

Set number of executors
--num-executors <number>

Set executor memory
--executor-memory <amount>

Set executor cores
--executor-cores <number>

Set driver memory
--driver-memory <amount>

Enable dynamic allocation
--conf spark.dynamicAllocation.enabled=true

Runtime Metrics

Enable metrics collection
--conf spark.metrics.conf=<metrics properties file>

Log metrics to Graphite
--conf spark.metrics.sink.graphite.class=org.apache.spark.metrics.sink.GraphiteSink

Connect to JMX
--conf spark.metrics.conf=metrics.properties (with JMX settings)

Enable event logging
--conf spark.eventLog.enabled=true

Set event log directory
--conf spark.eventLog.dir=<directory>

SQL Commands

Create a temporary view
dataFrame.createOrReplaceTempView("viewName")

Run a SQL query
spark.sql("SELECT * FROM viewName")

Show tables
spark.sql("SHOW TABLES").show()

Describe table
spark.sql("DESCRIBE viewName").show()

Cache table
spark.sql("CACHE TABLE viewName").show()

Debugging

Set log level
sc.setLogLevel("ERROR")

Print DAG visualization
df.explain()

Get execution plan
spark.sql("EXPLAIN SELECT * FROM viewName").show()

Count records in RDD/DataFrame
df.count()

Show DataFrame schema
df.printSchema()