Apache Spark & Kafka Lab Manual

Big Data Processing & Real-Time Analytics

SECTION 1: SETUP & ENVIRONMENT

Quick Setup Script

```
cd ~
cat > spark_kafka_setup.sh << 'SETUPEOF'
#!/bin/bash

echo "Installing Spark & Kafka..."

# Update & Install Prerequisites
sudo apt update -qq && sudo apt upgrade -y -qq
sudo apt install -y wget curl vim openjdk-11-jdk python3 python3-pip python3-venv

# Set Java
echo 'export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64' >> ~/.bashrc
echo 'export PATH=$PATH:$JAVA_HOME/bin' >> ~/.bashrc

# Install Spark
cd ~
wget -q https://archive.apache.org/dist/spark/spark-3.5.0/spark-3.5.0-bin-hadoop3.tgz
tar -xzf spark-3.5.0-bin-hadoop3.tgz
sudo mv spark-3.5.0-bin-hadoop3 /opt/spark
rm spark-3.5.0-bin-hadoop3.tgz

echo 'export SPARK_HOME=/opt/spark' >> ~/.bashrc
echo 'export PATH=$PATH:$SPARK_HOME/bin:$SPARK_HOME/sbin' >> ~/.bashrc
echo 'export PYSPARK_PYTHON=python3' >> ~/.bashrc

# Configure Spark
cd /opt/spark/conf
sudo cp spark-env.sh.template spark-env.sh
sudo tee -a spark-env.sh > /dev/null << 'EOF'
export SPARK_MASTER_HOST='localhost'
export SPARK_MASTER_PORT=7077
export SPARK_WORKER_CORES=2
```

```
export SPARK_WORKER_MEMORY=4g
export SPARK_DRIVER_MEMORY=2g
EOF

# Install Kafka
cd ~
wget -q https://archive.apache.org/dist/kafka/3.6.0/kafka_2.12-3.6.0.tgz
tar -xzf kafka_2.12-3.6.0.tgz
sudo mv kafka_2.12-3.6.0 /opt/kafka
rm kafka_2.12-3.6.0.tgz

echo 'export KAFKA_HOME=/opt/kafka' >> ~/.bashrc
echo 'export PATH=$PATH:$KAFKA_HOME/bin' >> ~/.bashrc

# Configure Kafka
cd /opt/kafka/config
sudo tee zookeeper.properties > /dev/null << 'EOF'
dataDir=/tmp/zookeeper
clientPort=2181
maxClientCnxns=0
admin.enableServer=false
EOF

sudo tee server.properties > /dev/null << 'EOF'
broker.id=0
listeners=PLAINTEXT://localhost:9092
advertised.listeners=PLAINTEXT://localhost:9092
log.dirs=/tmp/kafka-logs
num.partitions=3
zookeeper.connect=localhost:2181
offsets.topic.replication.factor=1
EOF

# Python Environment
cd ~
mkdir -p spark-kafka-lab/data
cd spark-kafka-lab
python3 -m venv venv
source venv/bin/activate
pip install -q pyspark==3.5.0 kafka-python==2.0.2 pandas numpy faker flask flask-socketio

# ------------------------
# Download NYC Taxi Parquet Files (Jan–Mar 2023)
# ------------------------
```

```
mkdir -p ~/spark-kafka-lab/data
cd ~/spark-kafka-lab/data

echo "Downloading NYC taxi Parquet files (Jan–Mar 2023)..."
wget -q https://d37ci6vzurychx.cloudfront.net/trip-data/yellow_tripdata_2023-01.parquet -O
yellow_tripdata_2023-01.parquet || echo "Download failed for Jan"
wget -q https://d37ci6vzurychx.cloudfront.net/trip-data/yellow_tripdata_2023-02.parquet -O
yellow_tripdata_2023-02.parquet || echo "Download failed for Feb"
wget -q https://d37ci6vzurychx.cloudfront.net/trip-data/yellow_tripdata_2023-03.parquet -O
yellow_tripdata_2023-03.parquet || echo "Download failed for Mar"

echo "Files downloaded:"
ls -lh yellow_tripdata_2023-*.parquet || true

# Fallback: generate small parquet sample if none downloaded
PARQUET_COUNT=$(ls yellow_tripdata_2023-*.parquet 2>/dev/null | wc -l || true)
if [ "$PARQUET_COUNT" -lt 1 ]; then
  echo "Parquet download failed — generating a smaller sample (1,000,000 rows) as
data/large_dataset_sample.parquet"
  python3 - <<'PYEOF'
import pandas as pd, numpy as np, os
from datetime import datetime, timedelta
np.random.seed(42)
n = 1000000
data = {
  'trip_id': range(1, n+1),
  'pickup_datetime': [datetime(2023,1,1) +
timedelta(seconds=int(np.random.randint(0,86400*30))) for _ in range(n)],
  'dropoff_datetime': [datetime(2023,1,1) +
timedelta(seconds=int(np.random.randint(0,86400*30))) for _ in range(n)],
  'passenger_count': np.random.randint(1,7,n),
  'trip_distance': np.random.exponential(3,n),
  'fare_amount': np.random.uniform(5,100,n),
  'tip_amount': np.random.uniform(0,20,n),
  'total_amount': np.random.uniform(5,120,n),
  'payment_type': np.random.choice(['Credit','Cash','Mobile'],n)
}
df = pd.DataFrame(data)
os.makedirs('data', exist_ok=True)
df.to_parquet('data/large_dataset_sample.parquet', index=False)
print('Created data/large_dataset_sample.parquet (1,000,000 rows)')
PYEOF
fi
```

```
# Download Iris
cd data
wget -q https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data -O iris.csv
echo "sepal_length,sepal_width,petal_length,petal_width,species" | cat - iris.csv > temp && mv
temp iris.csv

# Start Services
source ~/.bashrc
export SPARK_HOME=/opt/spark
export KAFKA_HOME=/opt/kafka
export PATH=$PATH:$SPARK_HOME/bin:$SPARK_HOME/sbin:$KAFKA_HOME/bin

mkdir -p /tmp/spark-events
$SPARK_HOME/sbin/start-master.sh
sleep 3
$SPARK_HOME/sbin/start-worker.sh spark://localhost:7077
sleep 3
$KAFKA_HOME/bin/zookeeper-server-start.sh -daemon
$KAFKA_HOME/config/zookeeper.properties
sleep 5
$KAFKA_HOME/bin/kafka-server-start.sh -daemon $KAFKA_HOME/config/server.properties
sleep 5

# Create Topics
$KAFKA_HOME/bin/kafka-topics.sh --create --topic sensor-data --bootstrap-server
localhost:9092 --partitions 3 --replication-factor 1 --if-not-exists
$KAFKA_HOME/bin/kafka-topics.sh --create --topic transactions --bootstrap-server
localhost:9092 --partitions 3 --replication-factor 1 --if-not-exists
$KAFKA_HOME/bin/kafka-topics.sh --create --topic alerts --bootstrap-server localhost:9092 --
partitions 1 --replication-factor 1 --if-not-exists


echo "✓ Setup Complete!"
echo "Run: source ~/.bashrc && cd ~/spark-kafka-lab && source venv/bin/activate"
SETUPEOF

chmod +x spark_kafka_setup.sh
./spark_kafka_setup.sh
```

After setup:

```
source ~/.bashrc
cd ~/spark-kafka-lab
source venv/bin/activate
jps  # Verify: Master, Worker, QuorumPeerMain, Kafka
```