Application Development and Troubleshooting

The application was developed and tested on the following configuration prior to deployment on Openstack:

```
1. VM: Oracle VirtualBox 6.1
2. Linux OS: Mint Linux 20
3. RAM: 16GB
4. CPU: 2 logical processors
5. Disk Space: 50GB
```

IDE Development and Debugging

There is a property in the SSPSparkApp config.properties file called "run.ide" which needs to be set to true. You also have to remove the "provided" scope from the spark-core and spark-sql libraries in the Maven pom.xml for the ssp-spark-app module.

The effect of these changes allow the application to be run outside the deployed Spark cluster in "local" mode

There's no such issue running the SSPFlinkApp within the IDE

JetBrains IntelliJ IDE with JDK8 and the Scala plugin for Scala 11.8 is recommended for development.

Single Node All-In-One Deployment

A single node all-in-one Docker compose descriptor has been provided. It is possible to deploy this on an Ubuntu VM running on a laptop with the spec described above by:

```
docker-compose -f SSPProject/ssp-deployment/single-node/docker-compose.yml -d
```

Spark deployment troubleshooting

If there are issues submitting the application to Spark you will see some output from the spark-submit command like this:

```
log4j:WARN No appenders could be found for logger (org.apache.hadoop.util.Nativ
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more i
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
20/04/18 18:32:34 INFO SecurityManager: Changing view acls to: spark
20/04/18 18:32:34 INFO SecurityManager: Changing modify acls to: spark
20/04/18 18:32:34 INFO SecurityManager: Changing view acls groups to:
20/04/18 18:32:34 INFO SecurityManager: Changing modify acls groups to:
20/04/18 18:32:34 INFO SecurityManager: SecurityManager: authentication disable
```

```
20/04/18 18:32:34 INFO Utils: Successfully started service 'driverClient' on po
20/04/18 18:32:34 INFO TransportClientFactory: Successfully created connection
20/04/18 18:32:34 INFO ClientEndpoint: Driver successfully submitted as driver-
20/04/18 18:32:34 INFO ClientEndpoint: ... waiting before polling master for dr
20/04/18 18:32:39 INFO ClientEndpoint: ... polling master for driver state
20/04/18 18:32:39 INFO ClientEndpoint: State of **driver-20200418183234-0001**
20/04/18 18:32:39 INFO ShutdownHookManager: Shutdown hook called
20/04/18 18:32:39 INFO ShutdownHookManager: Deleting directory /tmp/spark-8f3e3
```

If something like this occurs then check the following location:

```
docker exec -it openstack-three-node_spark-worker_1 bash
cd /opt/bitnami/spark/work/
ls -lrtd driver*
cd <last driver directory>
cat stderr
```

Spark logs

The Spark logs can be checked as follows:

```
docker exec -it openstack-three-node_spark-master_1 bash
cd /opt/bitnami/spark/work/
ls -lrtd driver*
cd <last driver directory>
cat stderr
```

The logs are also viewable via the Spark UI : http://localhost:8080

Configuration Options

SSPDataImport (ssp-data-import/src/main/resources/config.properties)

```
# AWS S3 connection properties (Data Source)
aws.bucketname = x19139497
aws.object.prefix = Telecommunications - SMS, Call, Internet - TN/sms-call-inte

# Kafka connection properties (Data Sink)
kafka.persist = true
kafka.server = kafka:9092
kafka.topic = telecom_trento

# Elasticsearch connection properties (Optional ES storage of raw 10min aggrega
es.persist = false
```

```
es.server = elasticsearch
es.port = 9200
es.scheme = http
es.index = dataimportcdr
es.bulk.offset = 10000
```

SSPSparkApp (ssp-spark-app/src/main/resources/config.properties)

```
# DEBUG mode
# This is for running the Spark application within the IDE for debug purposes
# You also need to remove the "provided" scope from the spark-core and spark-s
# Also the /etc/hosts file on the host machine needs to be edited to add "kafk
# IDE run application to connect to the broker
run.ide = false

# Kafka connection properties (Data Source)
kafka.server = kafka:9092
kafka.topics = telecom_trento
kafka.topic.starting.offset = earliest
kafka.max.offsets.per.trigger = 1000000

# Elasticsearch connection properties (Data Sink)
es.server = elasticsearch
es.port = 9200
es.scheme = http
es.index = sparkcdr

# Aggregation Window settings (Tumbling Window)
time.window.secs = 60

# Aggregation time enablement
enable.hourly.agg = true
enable.daily.agg = true
enable.weekly.agg = true
```

SSPFlinkApp (ssp-flink-app/src/main/resources/config.properties)

```
# Kafka connection properties (Data Source)
kafka.server = kafka:9092
kafka.topics = telecom_trento
kafka.earliest.offset = true

# Elasticsearch connection properties (Data Sink)
es.server = elasticsearch
es.port = 9200
es.index = flinkcdr
```

```
# Aggregation Window settings (Tumbling Window)
time.window.secs = 60

# Aggregation time enablement
enable.hourly.agg = true
enable.daily.agg = true
enable.weekly.agg = true
```

Programming References

Docker Compose

```
https://docs.docker.com/compose/compose-file/
```

Wurtmeister Docker images for Kafka

```
http://wurstmeister.github.io/kafka-docker/
https://github.com/wurstmeister/kafka-docker/blob/master/README.md
https://github.com/wurstmeister/kafka-docker/wiki/Connectivity
```

Bitnami Docker images for Spark

```
https://github.com/bitnami/bitnami-docker-spark
```

Kakfa - Spark integration

```
https://spark.apache.org/docs/latest/structured-streaming-kafka-integration.htr
https://spark.apache.org/docs/latest/streaming-kafka-0-10-integration.html
```

Kakfa - Flink integration

```
https://ci.apache.org/projects/flink/flink-docs-stable/dev/connectors/kafka.htr
```

Elasticsearch Java High Level REST Client

```
https://www.elastic.co/guide/en/elasticsearch/client/java-rest/7.8/java-rest-h
```

Flink Elasticsearch Connector

```
https://ci.apache.org/projects/flink/flink-docs-stable/dev/connectors/elasticse
```

Kafka API

```
https://kafka.apache.org/documentation/
```