Table of Contents

ASPARC STARTER DEVOPS GUIDE

1. WAY OF WORKING

1.1. Issues handling

Each proper time spent on time saves 10 times more in execution, thus the tasks and aktivities related to this tool are tracked via the issue-tracker tool:
https://github.com/YordanGeorgiev/issue-tracker
and could be found @:
https://docs.google.com/spreadsheets/d/1-oYPtBM8PG_FUogk40RDmcM_Xzq91Tb81Zlyi0cMwYQ/edit#gid=135774576

1.2. Documentation

The purpose of the tool is to "grasp the concept of apache spark", thus a proper documentation set is created as well.

2. NAMING CONVENTIONS

2.1. Dirs naming conventions

The dir structure should be logical and a person navigating to a dir should almost understand what is to be find in thre by its name ..

2.2. Root Dirs naming conventions

The root dirs and named as follows:
bin - contains the produced binaries for th project
cnf - for the configuration
dat - for the data of the app
lib - for any external libraries used
src - for the source code of the actual projects and subprojects

2.3. Bash scripts naming conventions

Do not use capital letters - they are too noisy.

2.4. Scala code naming conventions

2.4.1. Configure java_home

Configure java_home env var to the the java_opts file.

```
echo 'export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64' >> ~/.java_opts.host-name
```

2.4.2. Verify the JDK installation and configuration

Verify the JDK installation and configuration as follows:

```
# and verify
java -version
java version "1.8.0_101"
Java(TM) SE Runtime Environment (build 1.8.0_101-b13)
```

```
Java HotSpot(TM) 64-Bit Server VM (build 25.101-b13, mixed mode)
```

2.5. Install Scala

The scala libs will be installed with the sbt build tool.

2.6. Install sbt

Install sbt scala build tool by following the instructions in the following url:
http://www.scala-sbt.org/0.13/docs/Installing-sbt-on-Linux.html

```
echo "deb https://dl.bintray.com/sbt/debian /" | sudo tee -a /etc/apt/sources.list.d/sbt.list
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 2EE0EA64E40A89B84B2DF73499E82A75642AC823
sudo apt-get update
sudo apt-get install sbt
which sbt
```

2.7. Install apache spark

2.7.1. Download the latest stable Apache Spak package

Download the spak package with curl as follows:

```
export dir=/vagrant/pckgs/apache
mkdir -p $dir ; cd $dir
curl -O https://d3kbcqa49mib13.cloudfront.net/spark-2.2.0-bin-hadoop2.7.tgz
```

2.7.2. Unpack and deploy

Download the spak package with curl as follows:

```
mv -v spark-2.2.0-bin-hadoop2.7/ spark
mv -v spark /usr/lib/
sudo mv -v spark /usr/lib/
```

2.7.3. Add env vars

Add the following env vars

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export SPARK_HOME=/usr/lib/spark
export PATH=$PATH:$SPARK_HOME

# reload the env vars
source ~/.profile_opts
```

2.7.4. Verify the installation

Verify the installation by startin the spark shell

```
spark-shell
```

```
Spark context available as 'sc' (master = local[*], app id = local-1505242880618).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 2.2.0
      /_/

Using Scala version 2.11.8 (OpenJDK 64-Bit Server VM, Java 1.8.0_131)
Type in expressions to have them evaluated.
Type :help for more information.

scala>
```

3. OPERATIONS

3.1. Run the examples

You can run all the examples as follows:

```
# check the actions to run
  cat src/bash/aspark-starter/tests/run-aspark-starter-tests.lst

# STDUOT
# sbt-compile-verbose
# sbt-clean-compile
# sbt-compile
# sbt-stage
# sbt-run

bash src/bash/aspark-starter/test-aspark-starter.sh

# now the tool will start producing output

# 2017-09-14 08:26:11      START test-aspark-starter test run report
#
# result  start-time  stop-time   action-name
#    ok    08:26:11 08:26:59 sbt-compile-verbose
#    ok    08:27:00 08:27:25 sbt-clean-compile
#    ok    08:27:25 08:27:34 sbt-compile
#    ok    08:27:35 08:27:49 sbt-stage
#    ok    08:27:49 08:27:59 sbt-run
```