Table of Contents

ASPARK-STARTER

1. WHAT IS IT ?!

A demo application which will help you to grasp the Apache Spark concept.

2. EASY DOCKER CONTAINER RUN

If you have docker on our System you could quickly check the functionality of the whole demo app by downloading the image from the public docker repository and running it in a container.

3. PULL THE IMAGE TO YOUR SYSTEM

Pull the image to your system as follows:

```
# pull the image with v4 tag from the 298fd226fcbc40d0b2a3a39258abc/aspark-starter docker repository
docker pull 298fd226fcbc40d0b2a3a39258abc/aspark-starter:v4
```

4. INSTANTIATE THE CONTAINER

List the images and start up a container.

```
# list he images after the download :
docker images  --all

# run the v3 image
docker run -itd fa2c3cede570 /bin/bash
```

5. START UP THE CONTAINER

Start up the container as follows:

```
# list he images after the download :
docker images  --all

# run the v3 image
docker run -itd fa2c3cede570 /bin/bash
```

6. BOOTSTRAP THE DEMO APP IN THE CONTAINER

The container will start with a different hostname compared to the oe which has been used for creating the image, thus you would have to change all the occurences of the parent container hostname to the current container host name as follows:

```
export to_srch=057230ffe0aa      # this was the parent container's hostname
export to_repl=`hostname -s`      # the hostname assigned for the instantiation
# the latest product instance dir where the demo app has been deployed
export dir_to_morph=/opt/csitea/aspark-starter/aspark-starter.0.1.2.dev.ysg
cd $dir_to_morph
# srch and replace both in file names and file paths
bash src/bash/aspark-starter/aspark-starter.sh -a morph-dir
```

7. RUN ALL THE EXAMPLE ACTIONS

The demo app is started by bash script running different actions. The run-local-app bash action has several scala actions which can be seen from the run-local-app.test.sh file.

```
# check which bash actions will be run
cat src/bash/aspark-starter/tests/run-aspark-starter-tests.lst

# and run them
bash src/bash/aspark-starter/test-aspark-starter.sh
```

8. INSTALLATION AND CONFIGURATION

This section presents the steps needed to perform to deploy the aspark-starter tool. Note, that the commands are for Ubuntu, thus if you are on different OS choose , google the names of the packages applicable for your OS.

8.1. Ensure you have the following prerequisite binaries

The must have binaries are:
bash, perl, zip
The nice to have are:
tmux, vim ,ctags
The examples are for Ubuntu - use you OS package manager …

```
apt-get autoclean
apt-get install --only-upgrade bash
sudo apt-get install -y perl
sudo apt-get install -y zip
apt-get upgrade
```

8.2. Fetch the source - option 1

Fetch the source from git hub as follows:

```
# create your product dir:
mkdir -p /opt/csitea/
cd /opt/csitea/

# fetch the source
git clone git@github.com:YordanGeorgiev/aspark-starter.git

# DO NOT CD into the new dir !!!!
```

8.3. Fetch the source - option 2

You could also fetch the source from the release page of GitHub even without git or GitHub configuration as follows:

```
# the example is for the 0.0.5 version - adapt to the most reasent one ...
wget https://github.com/YordanGeorgiev/aspark-starter/archive/0.0.5.zip
unzip -o 0.0.5.zip.1 -d .
```

```
bash aspark-starter-0.0.5/src/bash/aspark-starter/bootstrap-aspark-starter.sh
```

8.4. Clone the git repo

Clone the git repo as follows:

```
# this requires github restration and ssh keys setup in your github account ….
git clone git@github.com:YordanGeorgiev/aspark-starter.git
```

8.5. Ensure you have all the prerequisite binaries

Ensure you have all the prerequisite binaries by issuing the following command

```
# bootstrap the product instance dir
bash aspark-starter/src/bash/aspark-starter/install-prerequisites-for-aspark-starter-on-ubuntu.sh
```

8.6. Build the first aspark-starter instance

Build the aspark-starter instance by running the bootstrap script

```
# bootstrap the product instance dir
bash aspark-starter/src/bash/aspark-starter/bootstrap-aspark-starter.sh

# the script should prompt you to cd
```

8.7. Check the runnable actions

You could check the functions which could be run - aka "actions" by issuing the following command.

```
# check the runnable with the -a cmd arg actions
find . -name '*.func.sh' | sort
```

8.8. Run the examples

You can run all the examples by first checking which actions are configured for the next test run and perform the actual test run as follows:

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

# STDOUT
# 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
```

8.9. Start hacking

Start hacking … or wait check at least the test call running all the functions of the tool …

```
# opionally if you are in the vim camp open the "project relative files list file"
vim meta/.dev.aspark-starter

# Ctrl + Z , to put it on the backgound
# check the actions to test ( uncoment line in include in test run )
less src/bash/aspark-starter/tests/run-aspark-starter-tests.lst

# Ctrl + Z to put in the background
# Action  !!! - aka now run the tests
bash src/bash/aspark-starter/test-aspar-starter.sh
```

8.9.1. Build and compile

Build and compile

```
bash src/bash/aspark-starter/aspark-starter.sh -a sbt-compile
```

8.9.2. Run the example

Run the example

```
bash src/bash/aspark-starter/aspark-starter.sh -a run-local-app
```

9. PROJECT STATUS

You could track the advancement of the project from the following url:
https://docs.google.com/spreadsheets/d/e/2PACX-1vR0wo5N32EpubwxBfeFxi6X-eOmXwOPg4WSyA4qBSz1Yu0EyU34jl0xICgWzrFUSeEA_aC4RF7LRqx9/pubhtml
Note that the content on the url is updated on project actual status update (i.e. meaningful work or milestones & tasks compppletion)

10. ADDITIONAL DOCUMENTATION

You can find the full installation and operations instructions in the docs/md dir of the project:
The UserStories:
https://github.com/YordanGeorgiev/aspark-starter/blob/master/doc/md/aspark-starter-user-stories.md
The Requirements:
https://github.com/YordanGeorgiev/aspark-starter/blob/master/doc/md/aspark-starter-requirements.md
The DevOps Guide:
https://github.com/YordanGeorgiev/aspark-starter/blob/master/doc/md/aspark-starter-devops-guide.md
The Features and Functionalities Description:

https://github.com/YordanGeorgiev/aspark-starter/blob/master/doc/md/aspark-starter-features-and-functionalities.md
The same files could be retrieved in pdf format from the doc/pdf dir of the project.