

FCT Traineeship: work report



D. Castro

EP-SFT

<https://cern.ch/swan>



FCT
Fundação
para a Ciência
e a Tecnologia

Dec 12th, 2018

Follow-up Meeting of the Administrative Agreement between
FCT and CERN

Introduction



Introduction

- > Making the CERN notebook service a professional Big Data web analysis platform
 - EP-SFT
 - June 2017 – June 2018
- > Supervisors:
 - Enric Tejedor Saavedra
 - Danilo Piparo

SWAN



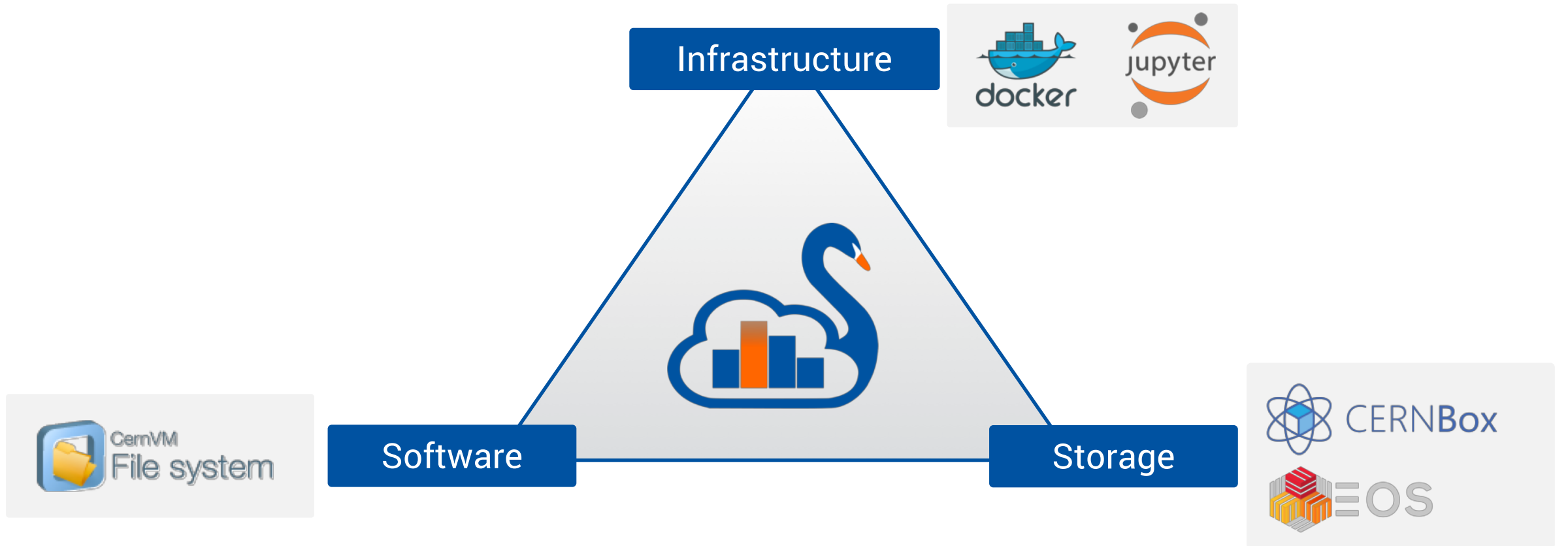
Service for Web based ANalysis

- > Sub project of the ROOT team
- > Analysis only with a web browser
 - Available everywhere and at anytime
 - Integrated with other analysis ecosystems : ROOT, R, Python, ...
 - No local installation and configuration needed
- > Leveraging the power of Jupyter notebooks
 - Create easily sharable scientific results: plots, data, code
- > Integration with CERN resources
 - Access software, user/experiments data, mass processing power






Integrator of services





Jupyter - The Notebook as Interface

Logout

SWAN Customisation

Specify the parameters that will be used to contextualise the container which is created for you. See [the online SWAN guide](#) for more details.


Software stack [more...](#)

Platform [more...](#)

Environment script [more...](#)

Number of cores [more...](#)

Memory [more...](#)

Control Panel Logout

Files [Running](#) [Clusters](#)

Select items to perform actions on them. Upload New ↻

	Name ↑	Last Modified ↑
	..	seconds ago
<input type="checkbox"/>	Proj1	5 days ago
<input type="checkbox"/>	Proj2	15 days ago
<input type="checkbox"/>	Project	21 days ago
<input type="checkbox"/>	Project 1	2 months ago
<input type="checkbox"/>	Project 2	4 months ago
<input type="checkbox"/>	Project 3	4 months ago
<input type="checkbox"/>	Project 4	4 months ago
<input type="checkbox"/>	Project 5	4 months ago
<input type="checkbox"/>	Project 6	4 months ago
<input type="checkbox"/>	ProjTest	15 days ago
<input type="checkbox"/>	Spark	7 days ago
<input type="checkbox"/>	Spark-Notebooks	14 days ago
<input type="checkbox"/>	SWAN-Spark_NXCALS_Example	20 days ago
<input type="checkbox"/>	Test Examples	a month ago
<input type="checkbox"/>	teste	19 days ago
<input type="checkbox"/>	SWAN-Spark_Simple_Example.ipynb	12 days ago
<input type="checkbox"/>	SWAN-Spark_Simple_Example.ipynb.orig	12 days ago



Simple_ROOTbook_py (unsaved changes) Control Panel Logout

File Edit View Insert Cell Kernel Help Trusted | Python 2

Code

Displaying graphics

We can now draw the histogram. We will at first create a [canvas](#), the entity which in ROOT holds graphics primitives. Note that thanks to [JSROOT](#), this is not a static plot but an interactive visualisation. Try to play with it and save it as image when you are satisfied!

```
In [5]: c = ROOT.TCanvas()  
h.Draw()  
c.Draw()
```

myHisto	
Entries	1000
Mean	0.02680
Std Dev	1.038

Text

Code

Graphics

My contribution to SWAN



My contribution to SWAN

> New User Interface

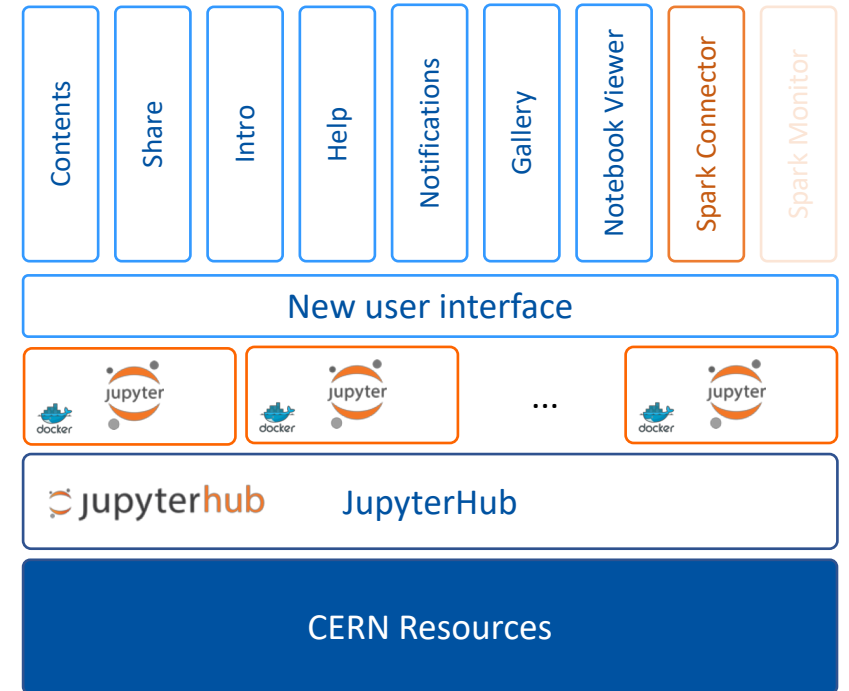
- Making it more usable and simple
- Many new functionalities

> New Sharing functionality

- Improving collaborative analysis

> Access to Spark clusters

- Simplifying and automating the connection to external computing resources



New User Interface

My contribution to SWAN



New User Interface

Configure Environment

Specify the parameters that will be used to contextualise the container which is created for you. See [the online SWAN guide](#) for more details.

Software stack [more...](#)
91

Platform [more...](#)
x86_64-slc6-gcc62-opt

Environment script [more...](#)
e.g. \$CERNBOX_HOME/MySWAN/myscript.sh

Number of cores [more...](#)
2

Memory [more...](#)
8 GB

Spark cluster [more...](#)
Hadalytic

Always start with this configuration

Start my Session

Starting your session

Waiting for swan-qa004.cern.ch...



New User Interface

SWAN > My Projects

My Projects

NAME	STATUS	MODIFIED
Proj1		5 days ago
Proj2		15 days ago
Project		21 days ago
Project 1		2 months ago
Project 2		4 months ago
ProjTest		15 days ago
Spark		7 days ago
SWAN-Spark_NXCALS_Example		20 days ago
teste		19 days ago

SWAN © Copyright CERN 2017. All rights reserved.
Home | Contacts | Support | Report a bug | Imprint

Spark > physics_analysis_using_swan_spark_template (autosaved)

FILE EDIT VIEW INSERT CELL KERNEL HELP Not Trusted Python 2

Integration of SWAN with Spark clusters

This notebook demonstrates the functionality provided by a SWAN prototype machine that allows to offload computations to an external Spark cluster. The Spark version we are going to use is 2.1.0 and we are going to connect to the analytix cluster (as previously selected in the SWAN web form).

Step 1 - Acquire the necessary credentials to access the Spark cluster.

```
In [1]: import getpass
import os, sys, re

print("Please enter your password")
ret = os.system("echo \"%s\" | kinit" % re.escape(getpass.getpass()))

if ret == 0: print("Credentials created successfully")
else:      sys.stderr.write('Error creating credentials, return code: %s\n' % ret)
```



Collaborative Analysis

My contribution to SWAN



Sharing made easy

- > Sharing from inside SWAN interface
 - Integration with CERNBox
- > Users can share “Projects”
 - Special kind of folder that contains notebooks and other files, like input data
 - Self contained

The screenshot displays the SWAN interface for sharing a project. The background shows a project titled "Super Real Analysis with TOTEM data" containing files like "DistillDistribution.ipynb" and "dataset.root". The foreground shows a "Share Project" dialog box with the following details:

- Share Project** (with a close button)
- You are sharing: **Super Real Analysis with TOTEM data**
- Search by name or username. Use "a:" for secondary accounts.
- Input field: Start typing to add names...
- Shared with**
- Users: Danilo Piparo (danilo), Enric Tejedor Saavedra (enric)
- Buttons: Stop Sharing, Update

SWAN © Copyright CERN 2016-2018. All rights reserved.
Home | Contact | Support | Report a bug





The Share tab

> Users can list which projects...

- they have shared
- others have shared with them

SWAN > Share

Projects shared with me ^

NAME ▾	SIZE	SHARED BY	DATE
UP2University Pilot	Empty	jupytercon2	7 minutes ago

Projects shared by me ^

NAME ▾	SHARED WITH	DATE
Higgs Boson discovery	2 people/groups	18 hours ago
Super Real Analysis with TOTEM data	diogo	19 hours ago

SWAN © Copyright CERN 2016-2018. All rights reserved.
Home | Contact | Support | Report a bug





Inspecting a Project

> Users can inspect shared project contents

- Browsing of the files
- Static rendering of notebooks

> Users can clone a shared Project

- Jupyter doesn't allow concurrent editing

Simple_ROOTbook_cpp.ipynb
(view-only)

Simple ROOTbook (C++)

This simple ROOTbook shows how to create a [histogram](#), [fill it](#) and [draw it](#). The language chosen is C++.

In order to activate the interactive visualisation we can use the [JSROOT](#) magic:

```
In [1]: %jsroot on
```

Now we will create a [histogram](#) specifying its title and axes titles:

```
In [2]: TH1F h("myHisto","My Histo;X axis;Y axis",64, -4, 4)
(TH1F &) Name: myHisto Title: My Histo NbinsX: 64
```

If you are wondering what this output represents, it is what we call a "printed value". The ROOT interpreter can indeed be instructed to "print" according to certain rules instances of a particular class.

Time to create a random generator and fill our histogram:

```
In [3]: TRandom3 rndmGenerator;
for (auto i : ROOT::TSeqI(1000)){
    auto rndm = rndmGenerator.Gaus();
    h.Fill(rndm);
}
```

We can now draw the histogram. We will at first create a [canvas](#), the entity which in ROOT holds graphics primitives.

```
In [4]: TCanvas c;
```

```
In [5]: h.Draw();
c.Draw();
```

my-Histo	
Entries	1000
Mean	0.02680
Std Dev	1.038

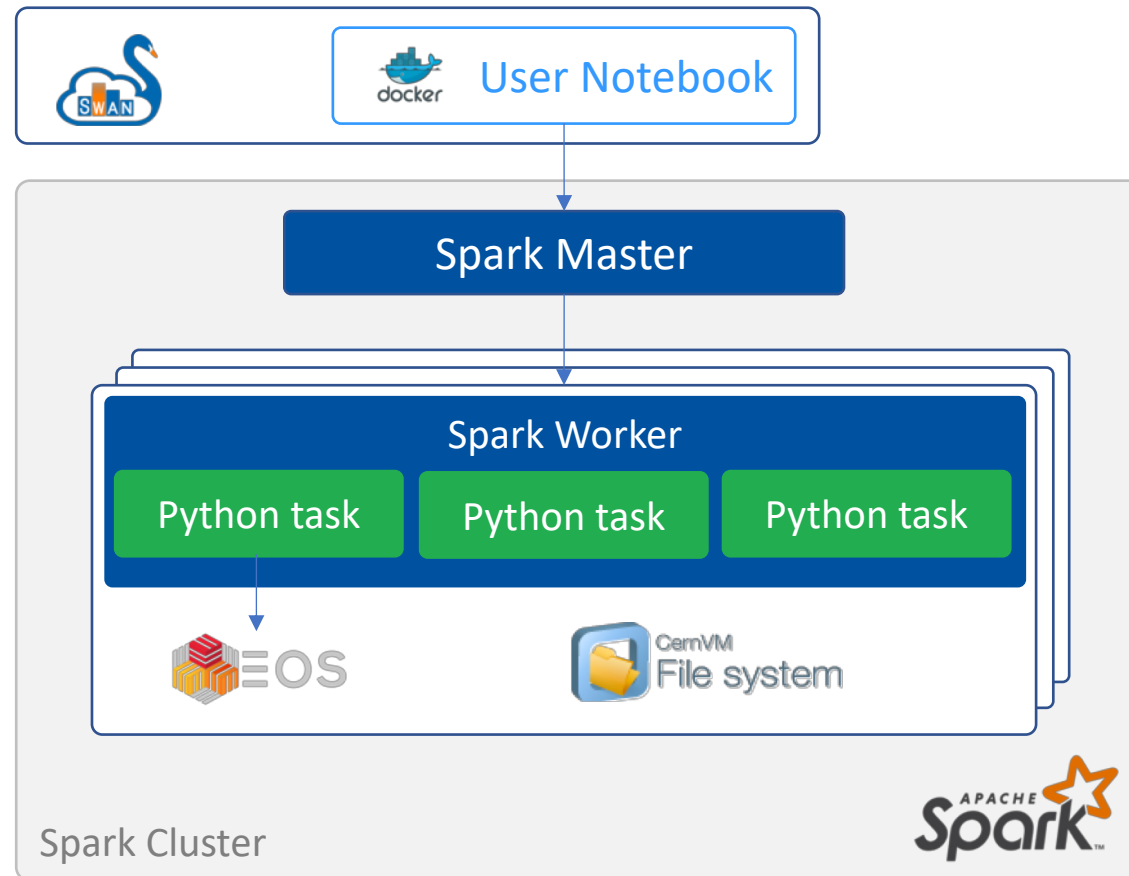
Access to Computing Resources

My contribution to SWAN



Integration with Spark

- > Connection to CERN Spark Clusters
- > Same environment across platforms
 - User data - EOS
 - Software - CVMFS
- > Graphical Jupyter extensions developed
 - Spark Connector
 - Spark Monitor





Spark Connector

Spark > Spark_Simple (autosaved)

FILE EDIT VIEW INSERT CELL KERNEL HELP

Spark clusters connection

You are going to connect to: **hadalytic**

You can configure the following [options](#). Environment variables can be used via {ENV_VAR_NAME}.

Add a new option

Write the option name...

Bundled configurations

Include NXCALs options

Selected configuration

- spark.shuffle.service.enabled false
- spark.driver.memory 2g
- spark.executor.instances 4

Connect

Spark > Spark_Simple Last Checkpoint: a minute ago (autosaved)

FILE EDIT VIEW INSERT CELL KERNEL HELP

Spark clusters connection

You are now connected

The following variables were instantiated:

- > **sc** = SparkContext
- > **spark** = SparkSession

Show/Hide connection logs

Go to the notebook



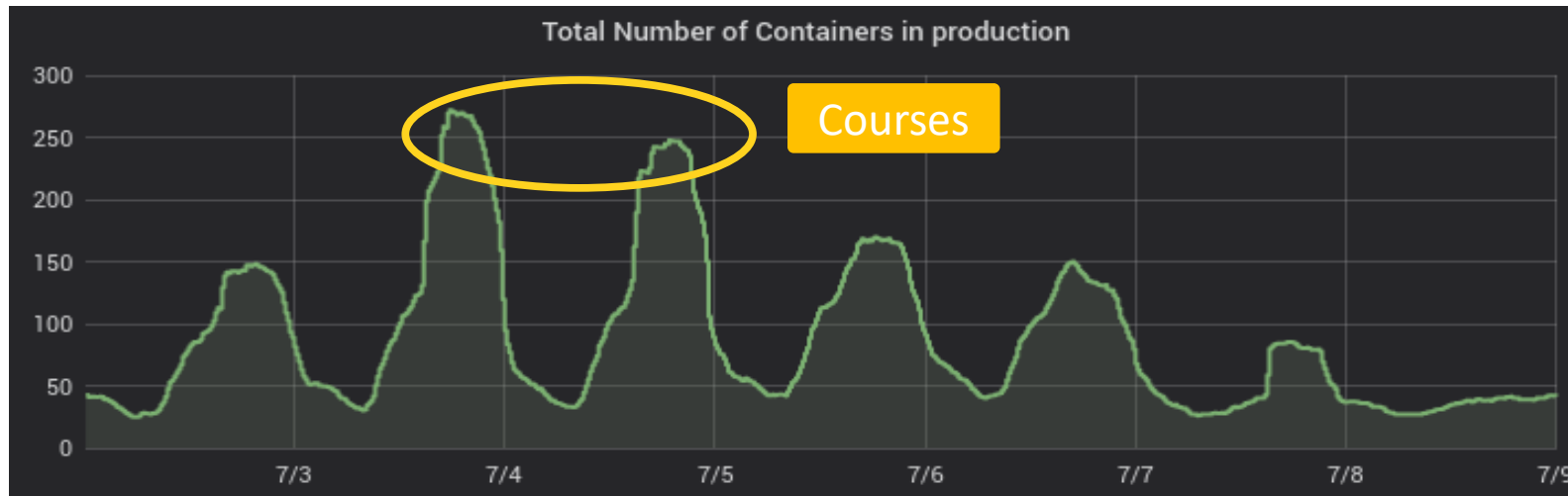
Impact on the service

My contribution to SWAN



Stats

- > **~200 user sessions** a day on average
 - Users doubled this year with new SWAN interface
- > Spark cluster connection: 15 – 20 % of users
 - SWAN as entry point for accessing computational resources



Other activities



Other activities

- > Service administration and support
- > Training
 - Tools for IT Service Managers: Getting started
 - Web Usability
 - Advanced Security
 - Python: Advanced Hands-On



Other activities

> GSoC mentoring

- Spark Monitor
- PyRDataFrame
- Batch jobs integration

> Presentations

- Several internal presentations
- CS3 2018 in Krakow
- Jupytercon 2018 in New York



Google Summer of Code



The move to IT-ST-FDO



> UP2University European Project

- Bridge the gap between secondary schools, higher education and the research domain
- Partner universities (OU, UROMA, NTUA, ...), pilot schools
- <http://up2university.eu>

> SWAN used by students to learn physics and other sciences

- Let them use the very same tools & services used by scientists at CERN
- Integrating Jupyter in teaching environment
- Complete SWAN package, with CERNBox, EOS and CVMFS



FCT Traineeship: work report

Thank you

Diogo Castro
diogo.castro@cern.ch

