SWAN: interactive data analysis on the web

Prasanth KOTHURI, IT-DB
On behalf of the SWAN team

https://cern.ch/swan

Sep 18th, 2019
PE Mini Lectures

Introduction

SWAN in a Nutshell

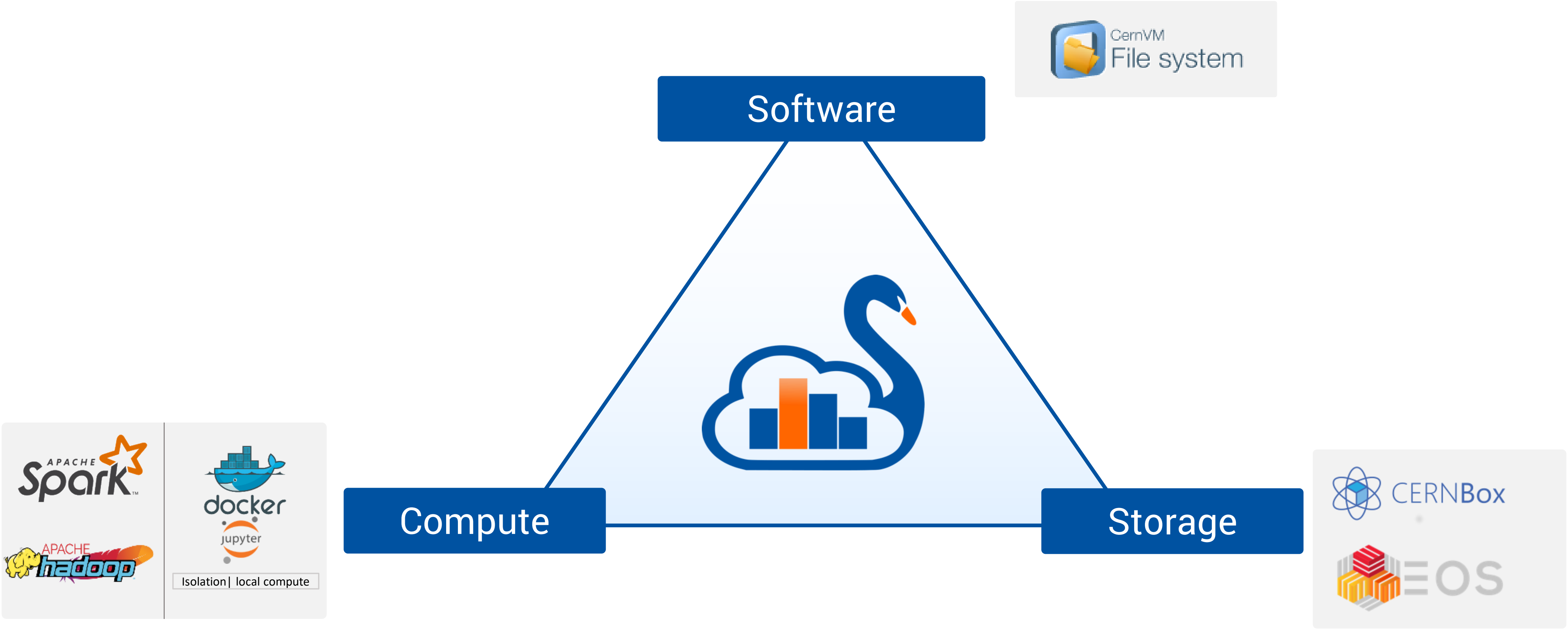
- Analysis only with a web browser
  - No local installation needed
  - Based on Jupyter Notebooks
  - Calculations, input data and results “in the Cloud”

- Support for multiple analysis ecosystems and languages
  - Python, ROOT C++, R and Octave

- Easy sharing of scientific results: plots, data, code

- Integration with CERN resources
  - Software, storage, mass processing power

Integrating services



Software

Compute

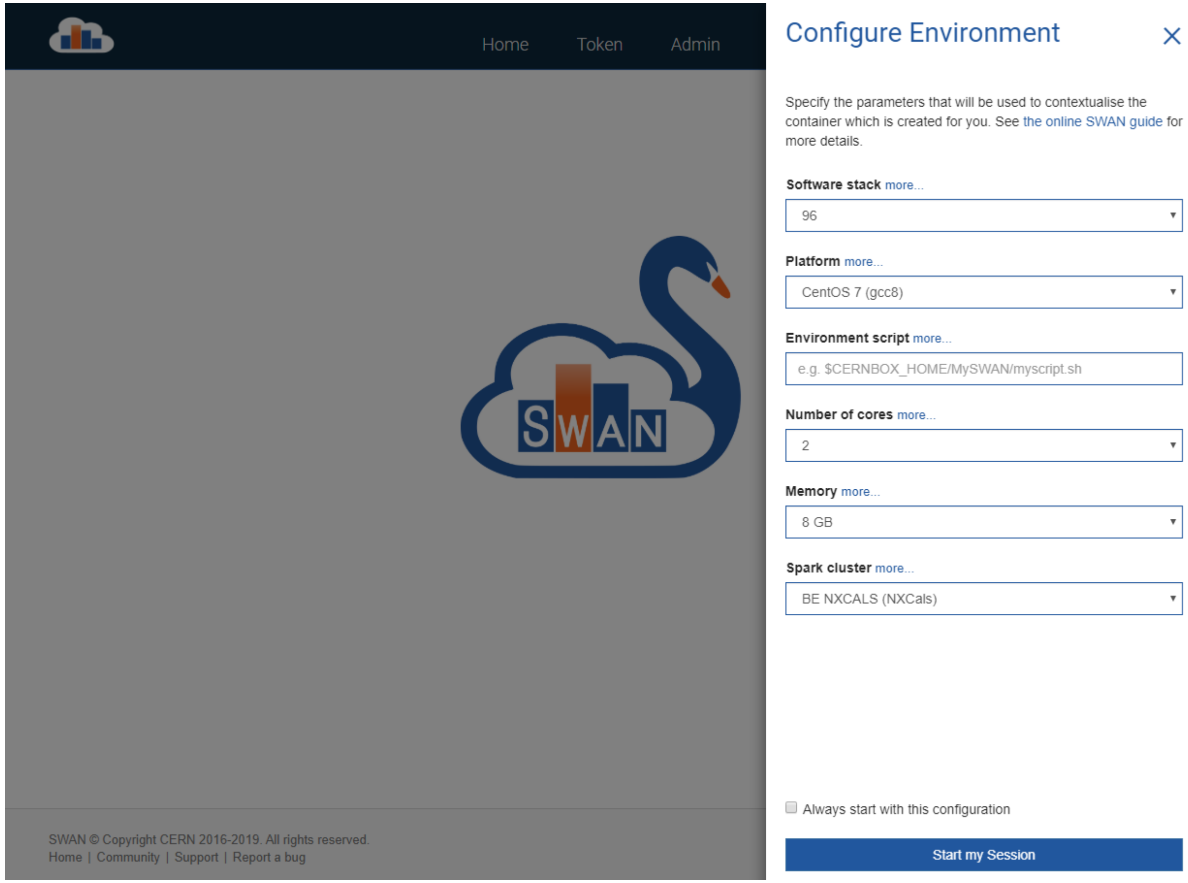
Storage

Isolation| local compute

Jupyter - The Notebook as Interface

- A web-based interactive interface and platform that combines code, equations, text and visualizations
  - Ideal for sharing/collaboration
  - A “shell opened within the browser”
- Interactive, usually lightweight computations and distributed parallel processing capability with the integration of mass processing system (Apache Spark)
- Very useful for some use cases
  - Final steps of an analysis, Exploration, Teaching, Documentation and Reproducibility

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...
96

Platform more...
CentOS 7 (gcc8)

Environment script more...
e.g. $CERNBOX_HOME/MySWAN/myscript.sh

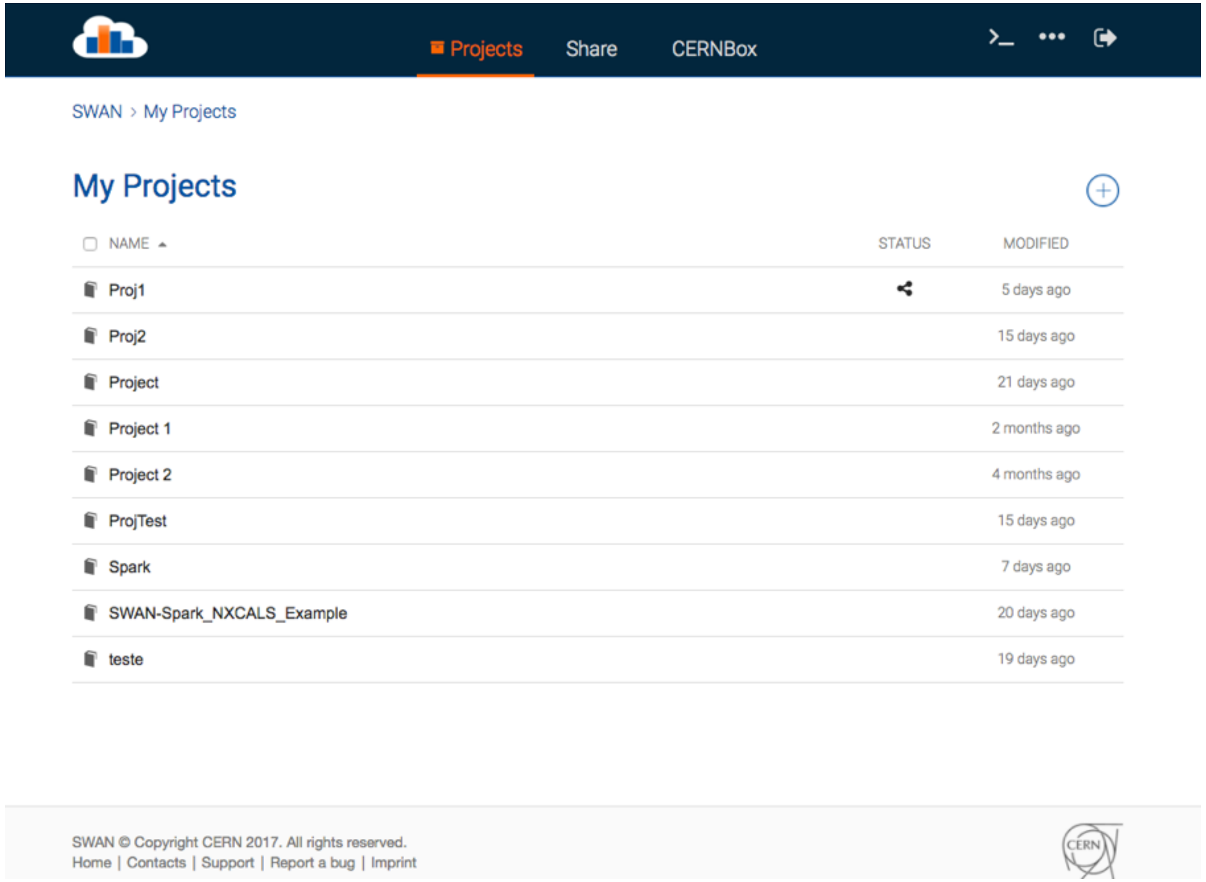
Number of cores more...
2

Memory more...
8 GB

Spark cluster more...
BE NXCALS (NXCals)

□ Always start with this configuration

SWAN © Copyright CERN 2016-2019. All rights reserved.
Home | Community | Support | Report a bug



My Projects

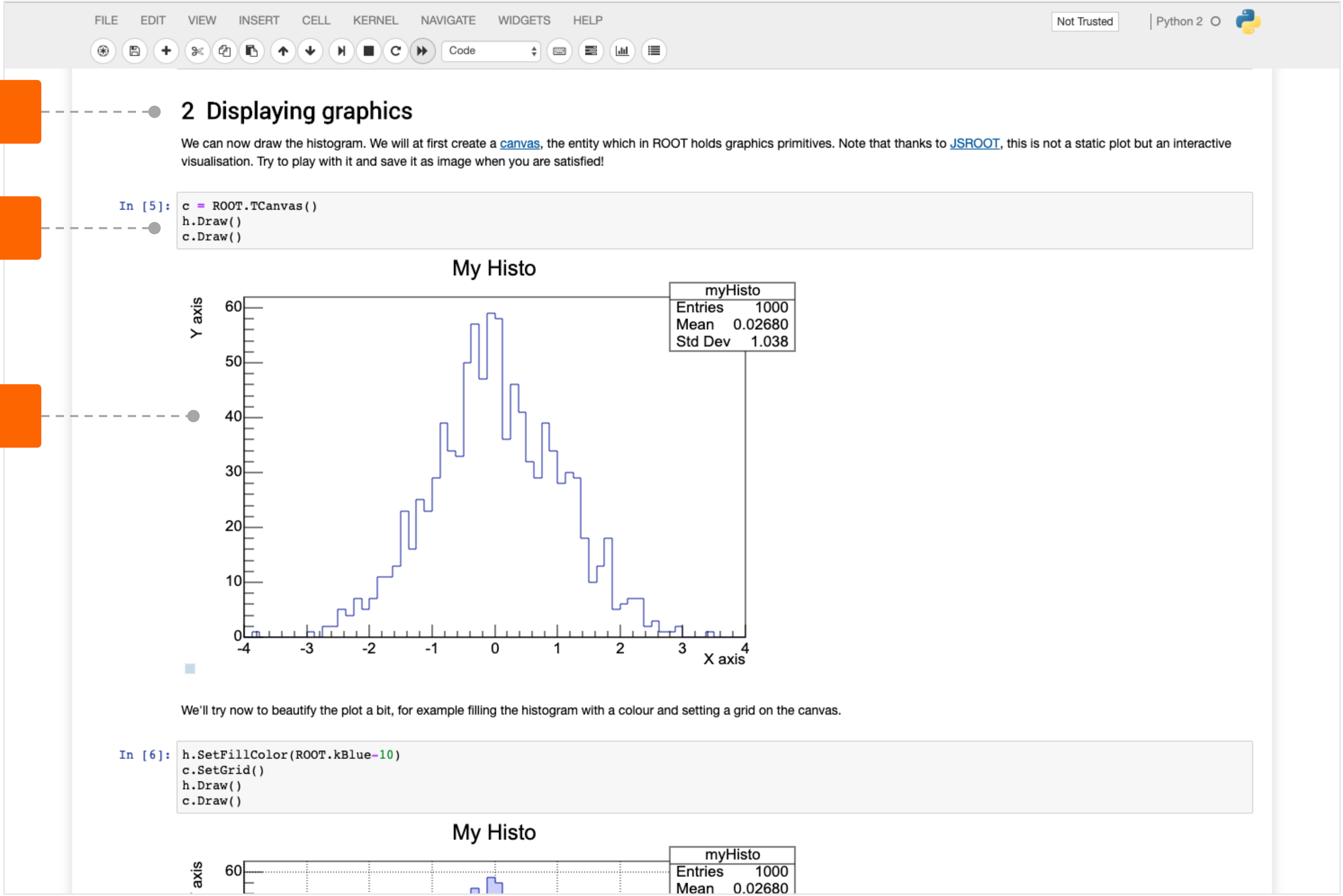
□ NAME ▲	STATUS	MODIFIED
Proj1	<	5 days ago
Proj2		15 days ago
Project		21 days ago
Project 1		2 months ago
Project 2		4 months ago
ProjTest		15 days ago
Spark		7 days ago
SWAN-Spark_NXCALS_Example		20 days ago
teste		19 days ago

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

Text

Code

Graphics



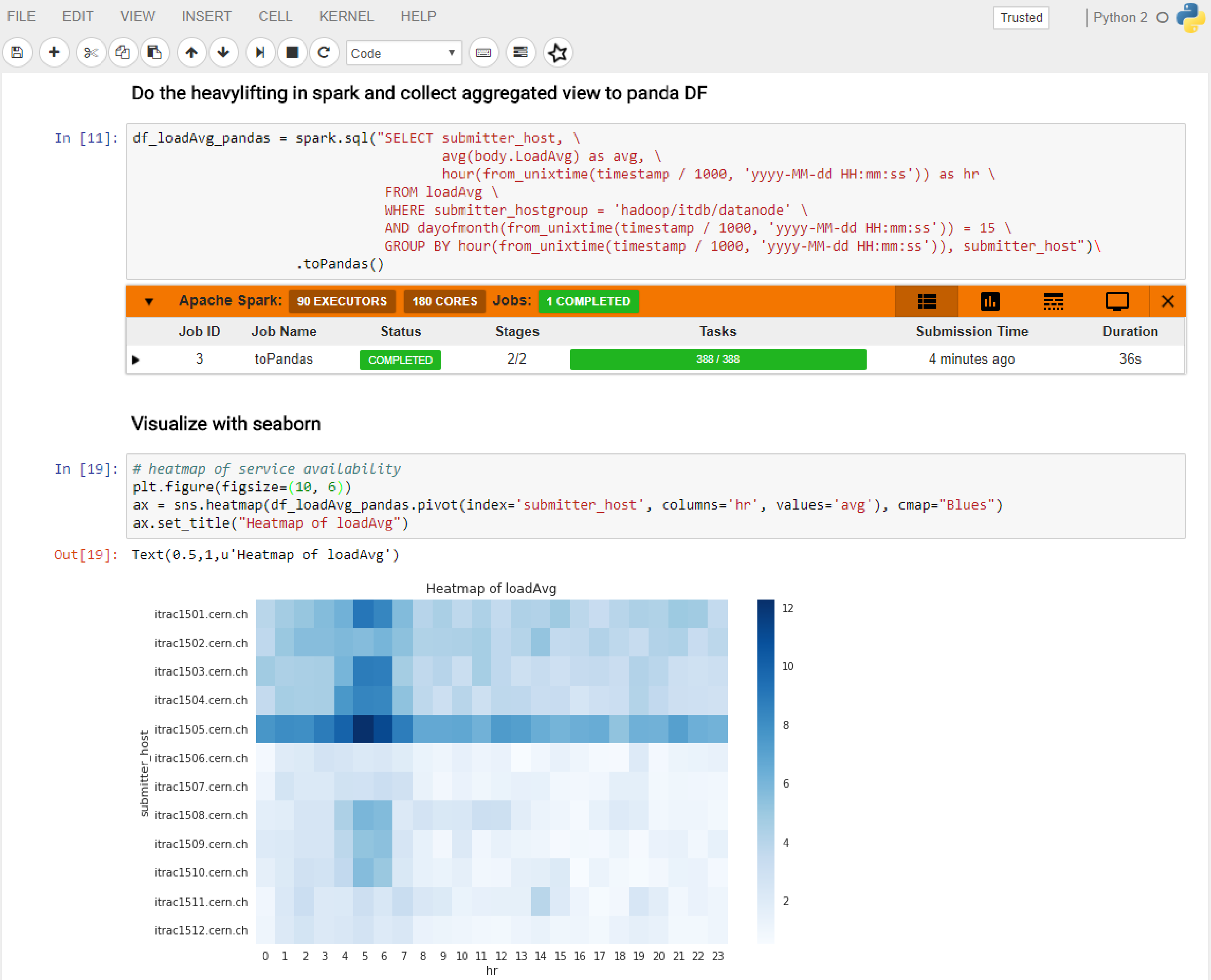
2 Displaying graphics

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()
```

We'll try now to beautify the plot a bit, for example filling the histogram with a colour and setting a grid on the canvas.

In [6]:
```
h.SetFillColor(ROOT.kBlue-10)
c.SetGrid()
h.Draw()
c.Draw()
```



Do the heavylifting in spark and collect aggregated view to panda DF

```
In [11]: df_loadAvg_pandas = spark.sql("SELECT submitter_host, \
                                          avg(body.LoadAvg) as avg, \
                                          hour(from_unixtime(timestamp / 1000, 'yyyy-MM-dd HH:mm:ss')) as hr \
                                   FROM loadAvg \
                                   WHERE submitter_hostgroup = 'hadoop/itdb/datanode' \
                                   AND dayofmonth(from_unixtime(timestamp / 1000, 'yyyy-MM-dd HH:mm:ss')) = 15 \
                                   GROUP BY hour(from_unixtime(timestamp / 1000, 'yyyy-MM-dd HH:mm:ss')), submitter_host")\
                            .toPandas()
```

Apache Spark: 90 EXECUTORS 180 CORES Jobs: 1 COMPLETED

Job ID	Job Name	Status	Stages	Tasks	Submission Time	Duration
3	toPandas	COMPLETED	2/2	388 / 388	4 minutes ago	36s

Visualize with seaborn

```
In [19]: # heatmap of service availability
         plt.figure(figsize=(10, 6))
         ax = sns.heatmap(df_loadAvg_pandas.pivot(index='submitter_host', columns='hr', values='avg'), cmap="Blues")
         ax.set_title("Heatmap of loadAvg")

Out[19]: Text(0.5,1,u'Heatmap of loadAvg')
```

Text

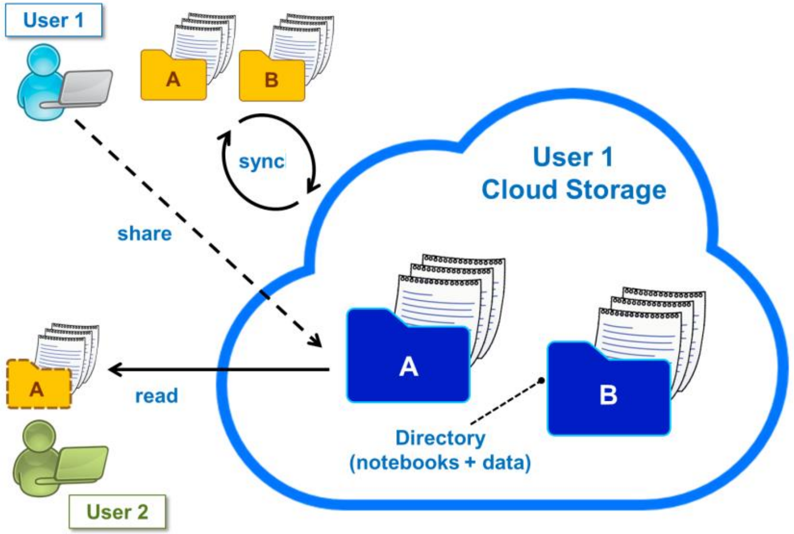
Code

Monitoring

Visualizations

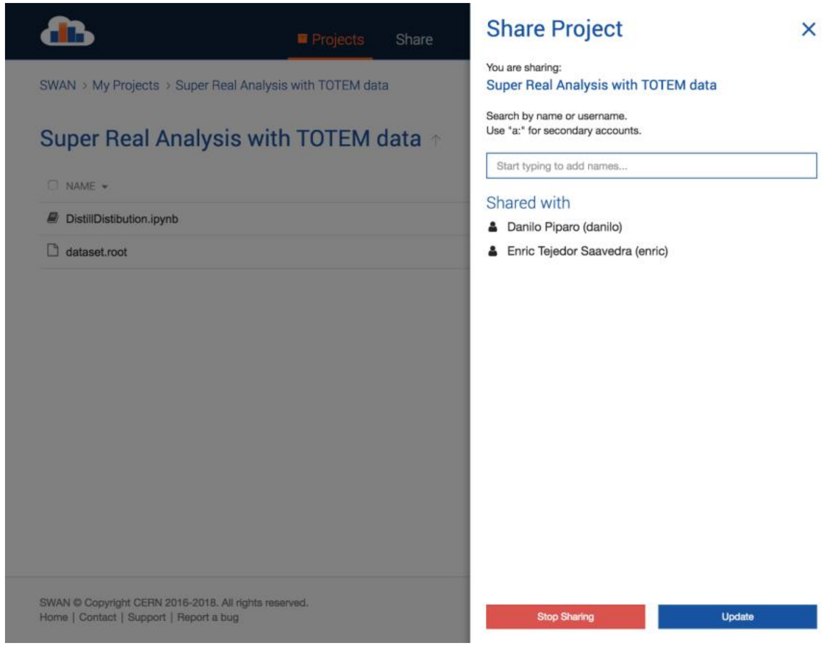
Cloud storage as your Home

- CERNBox is SWAN's home directory
  - Storage for your notebooks and data
  - Based on ownCloud
  - 6PB of user data, 16k users
- Uses EOS disk storage system
  - All experiment data potentially available
  - 250PB of experimental data at CERN (LHC and others)
- Sync&Share
  - Files synced across devices and the Cloud
  - Collaborative analysis



Sharing made easy

- Sharing from inside SWAN interface
  - Integration with CERNBox
  - List shares from other users
- Users can share “Projects”
  - Special kind of folder that contains notebooks and other files, like input data
  - Self contained
- Concurrent editing not supported *yet* by Jupyter
  - Safer to clone
  - Will be available with Jupyterlab



SWAN › My Projects › Super Real Analysis with TOTEM data

Super Real Analysis with TOTEM data

□ NAME

DistillDistibution.ipynb

dataset.root

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

Share Project

You are sharing:
Super Real Analysis with TOTEM data

Search by name or username.
Use "a:" for secondary accounts.

Start typing to add names...

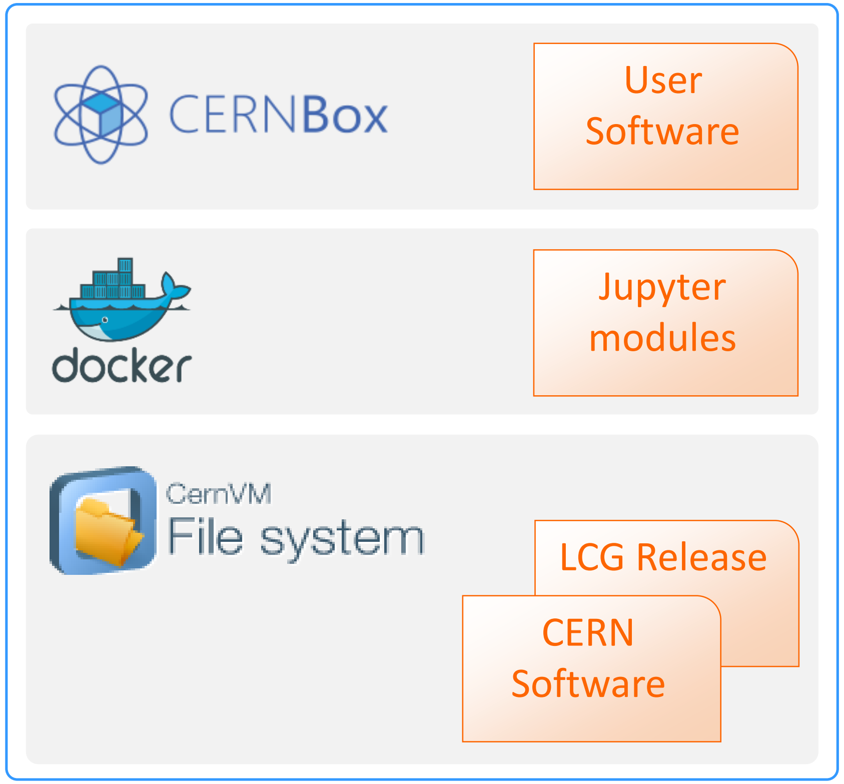
Shared with

Danilo Piparo (danilo)

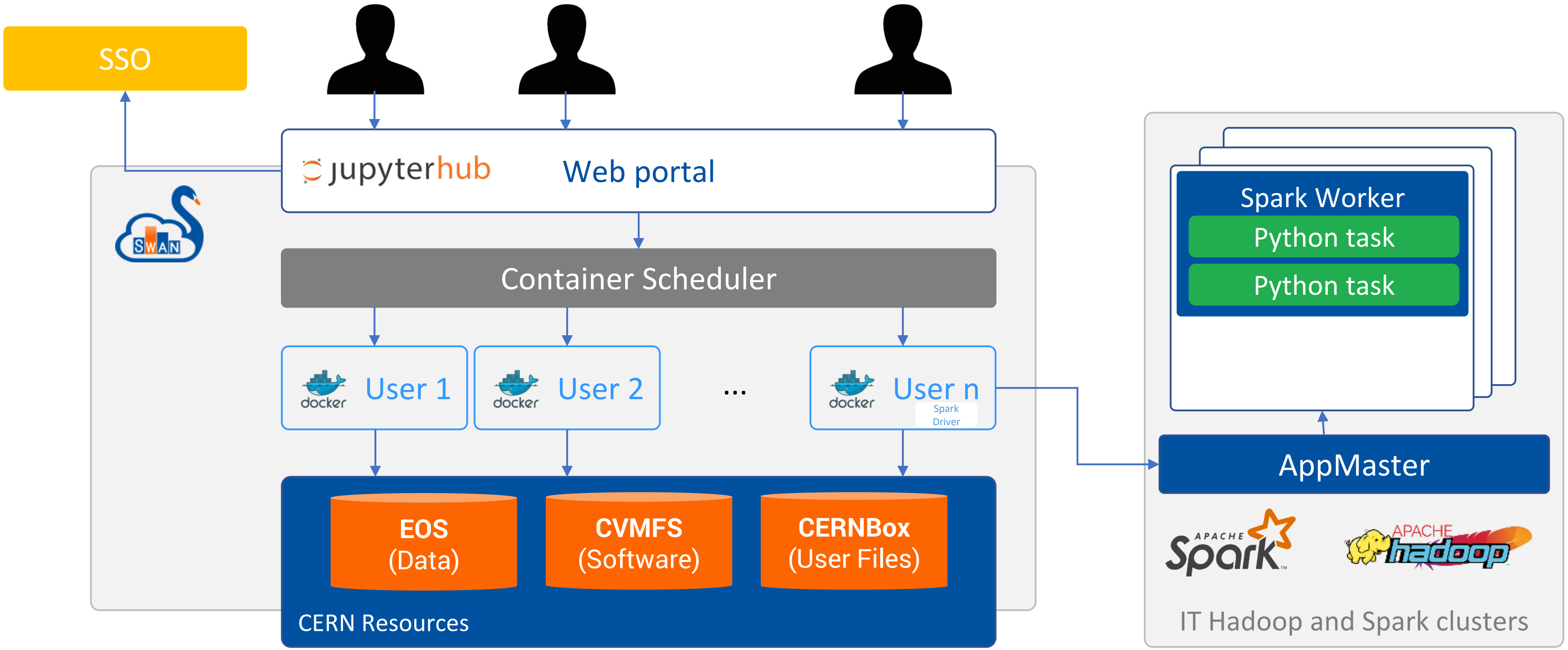
Enric Tejedor Saavedra (enric)

Software

- Software distributed through CVMFS
  - Distributed read-only filesystem
  - ”LCG Releases” - pack a series of compatible packages
  - Reduced Docker Images size
  - Lazy fetching of software
  - Step towards reproducibility (across time and people)
- Possibility to install libraries in user cloud storage
  - Good way to use custom/not mainstream packages
  - Configurable environment



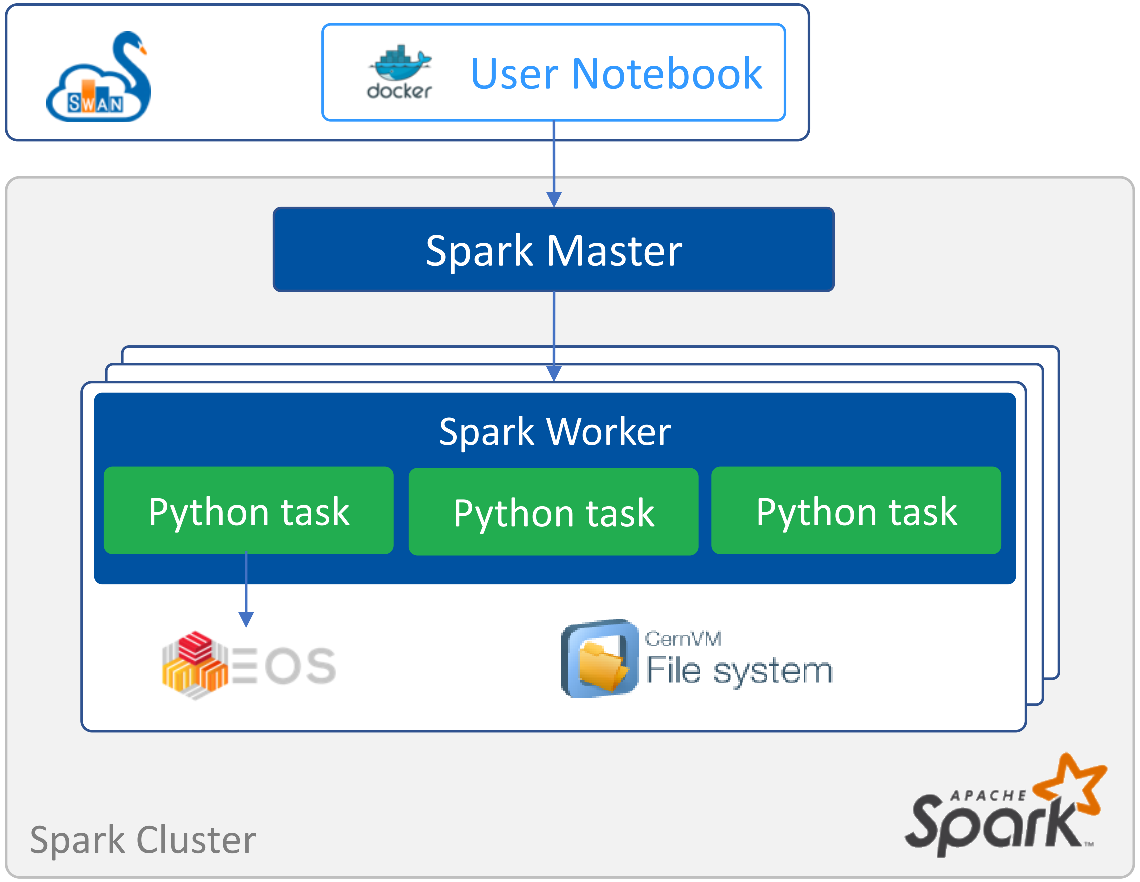
Architecture



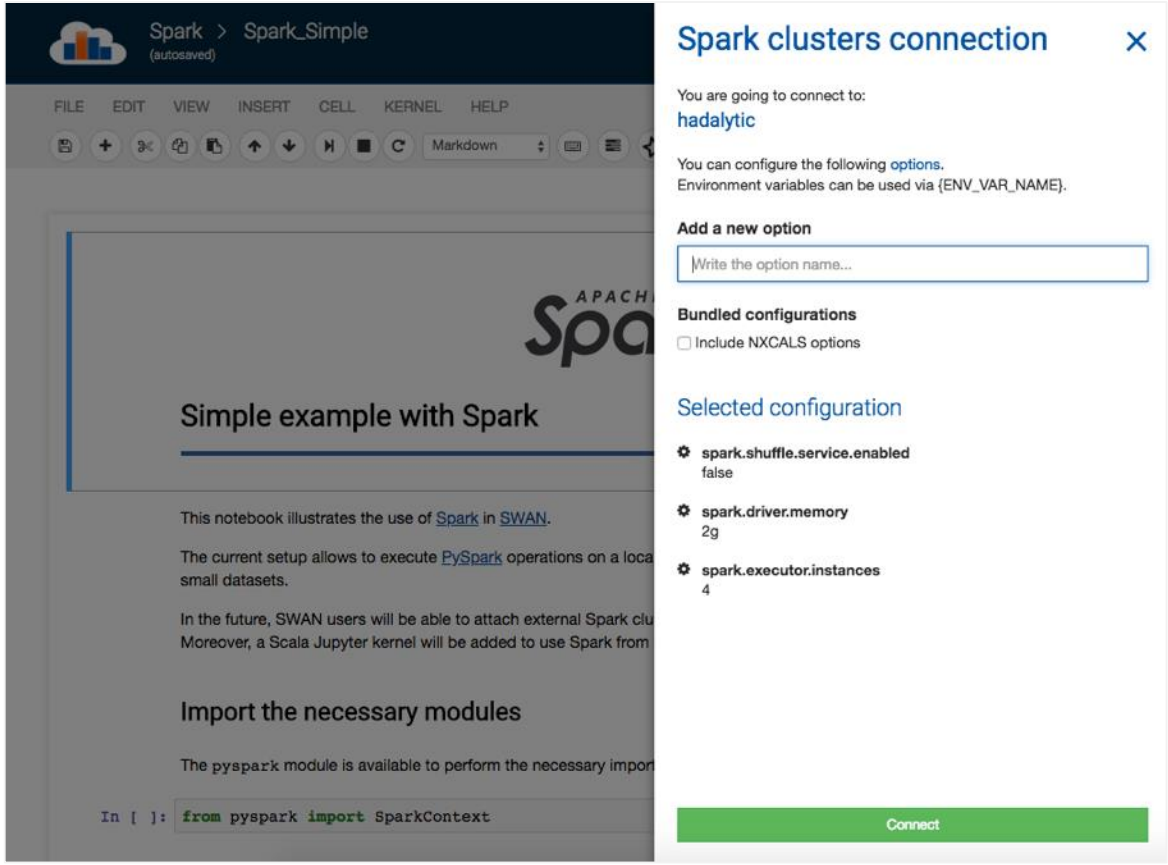
Access to Computing Resources

Integration with Spark

- Connection to CERN Spark Clusters
  - Spark: general purpose distributed computing framework
- Same environment across platforms (local/remote)
  - User data - EOS
  - Software - CVMFS
- Graphical Jupyter extensions developed
  - Spark Connector
  - Spark Monitor
- Not only used for Physics Analysis at CERN
- Spark Clusters
  - nxcals – Dedicated cluster for accelerator logging
  - Analytix – General purpose YARN cluster
  - Cloud Containers – General purpose Kubernetes cluster



Spark Connector



Simple example with Spark

This notebook illustrates the use of Spark in SWAN.

The current setup allows to execute PySpark operations on a loca
small datasets.

In the future, SWAN users will be able to attach external Spark clu
Moreover, a Scala Jupyter kernel will be added to use Spark from

Import the necessary modules

The pyspark module is available to perform the necessary impor

```
In [ ]: from pyspark import SparkContext
```

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

- Spark Connector – handling the spark configuration complexity
  - User is presented with Spark Session (Spark) and Spark Context (sc)
  - Ability to bundle configurations specific to user communities
  - Ability to specify additional configuration

A configurable system

Science Box: SWAN on Premises

- Packaged deployment of SWAN
  - Includes all SWAN components: CERNBox/EOS, CVMFS, JupyterHub
- Deployable through Kubernetes or docker-compose
- Some successful community installations…
  - AARNet
  - PSNC
  - Open Telekom Cloud (Helix Nebula)
- …with different storage integrations
  - OwnCloud WebDav access (AARNet)

Looking ahead

Future work/challenges

- Move to Jupyterlab
  - Porting the current extensions
  - Concurrent editing
- New architecture
  - Based on Kubernetes
- Exploitation of GPUs
  - HEP is looking to ML
  - Speed up computation of GPU-ready libraries (e.g. TensorFlow)

Demo

Where to find us

Where to find us

- Contacts
  - swan-admins@cern.ch
  - http://cern.ch/swan
- Repository
  - https://github.com/swan-cern/
- Science Box
  - https://cern.ch/sciencebox

Conclusion

Conclusion

- SWAN is a CERN service that provides Jupyter Notebooks on demand
  - Promotes a cloud-based analysis model
  - The new Jupyterlab interface will bring new possibilities for collaborative analysis with the introduction of concurrent editing of notebooks

- SWAN became a fundamental Interface for Mass Processing Resources (Spark) at CERN
  - Not only for Physics analysis but also for monitoring the LHC hardware

- Successfully deployed outside of CERN premises
  - Personalized to fit the local infrastructure
  - Ongoing effort to allow interoperability (CS3APIs, Science Mesh)

SWAN and its analysis ecosystem

Thank you

Prasanth KOTHURI
prasanth.kothuri@cern.ch