SWAN: interactive data analysis on the web

Prasanth Kothuri
On behalf of the SWAN team

https://cern.ch/swan

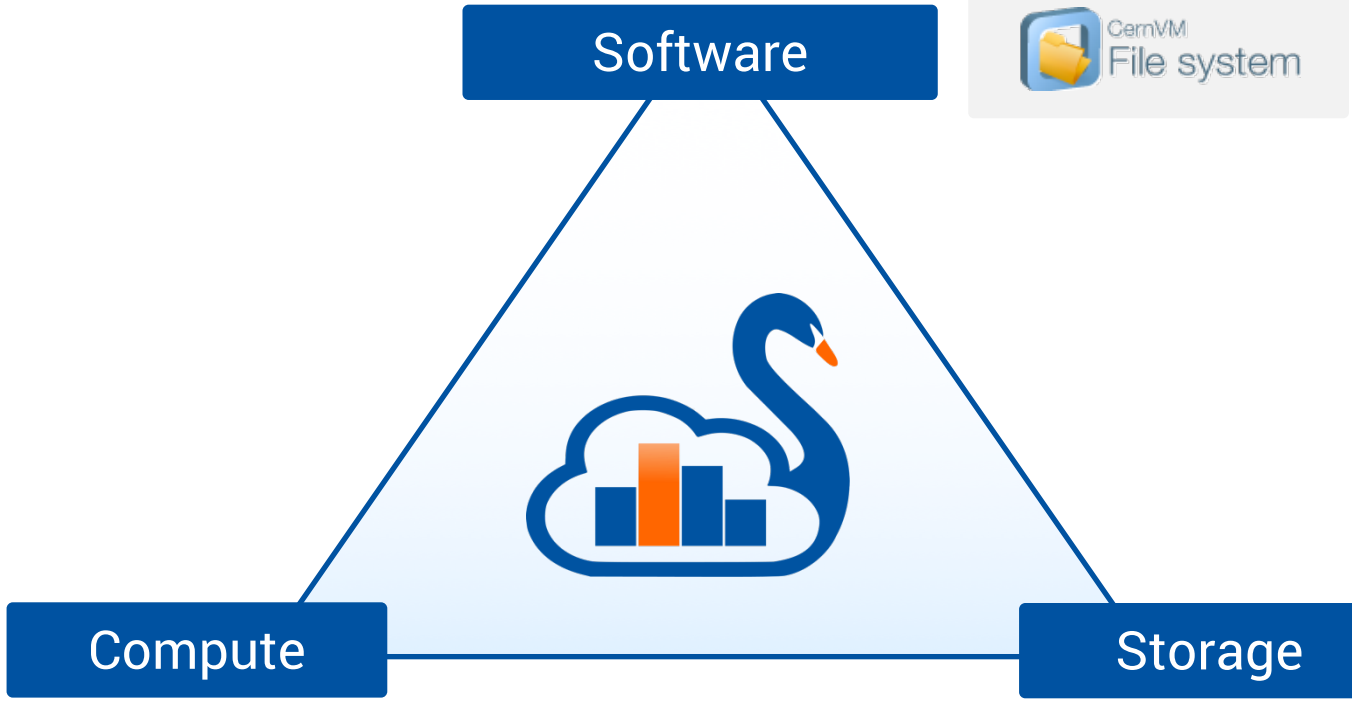
Oct 11th, 2019
SWAN Users' Workshop

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



CernVM File system

Spark, hadoop, docker, jupyter

Isolation| local compute

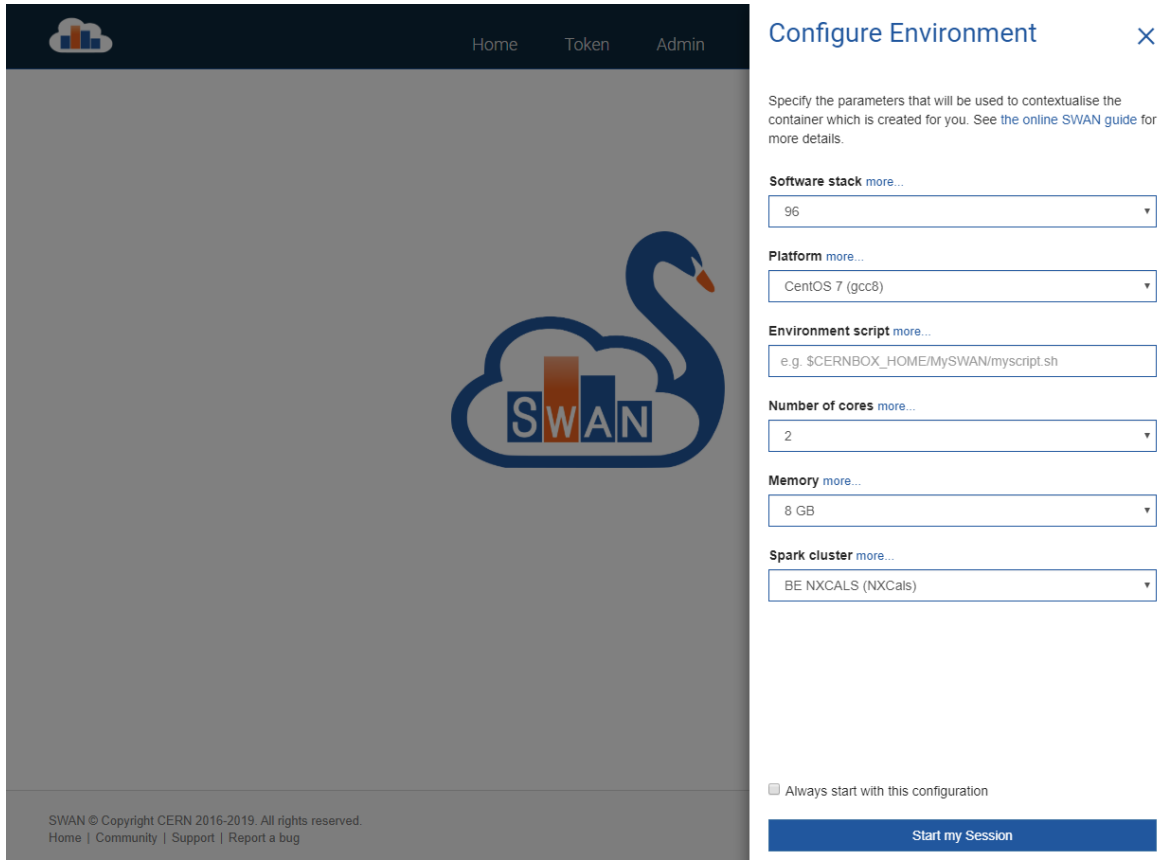
CERNBox

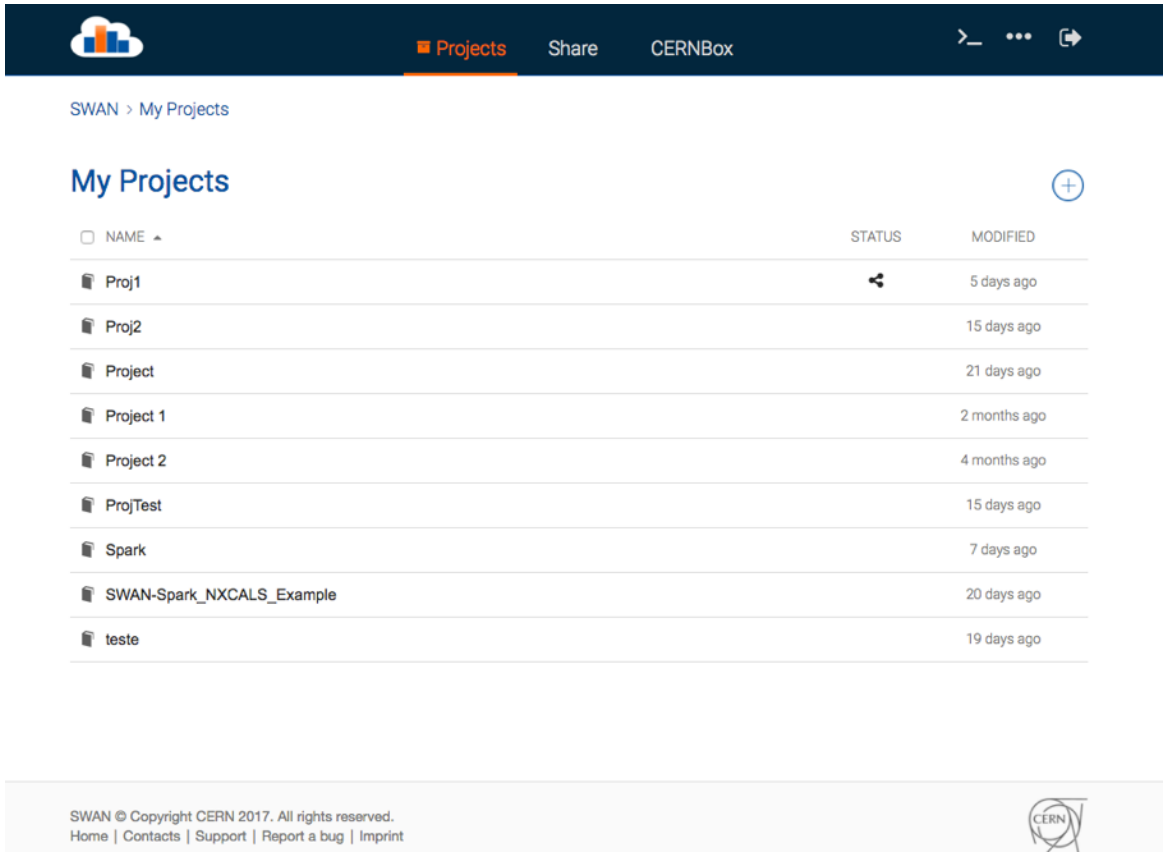
EOS

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 at CERN
  - Final steps of an Analysis, Exploration, Teaching, Documentation and Reproducibility

User Interface





Text

Code

Graphics

FILE EDIT VIEW INSERT CELL KERNEL NAVIGATE WIDGETS HELP

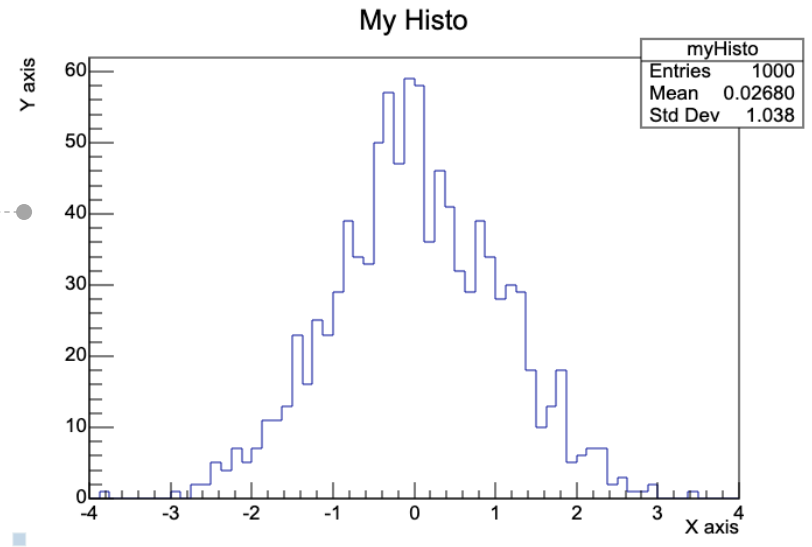
Not Trusted | Python 2

Code

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()
```

FILE EDIT VIEW INSERT CELL KERNEL HELP Trusted Python 2

Code

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()
```

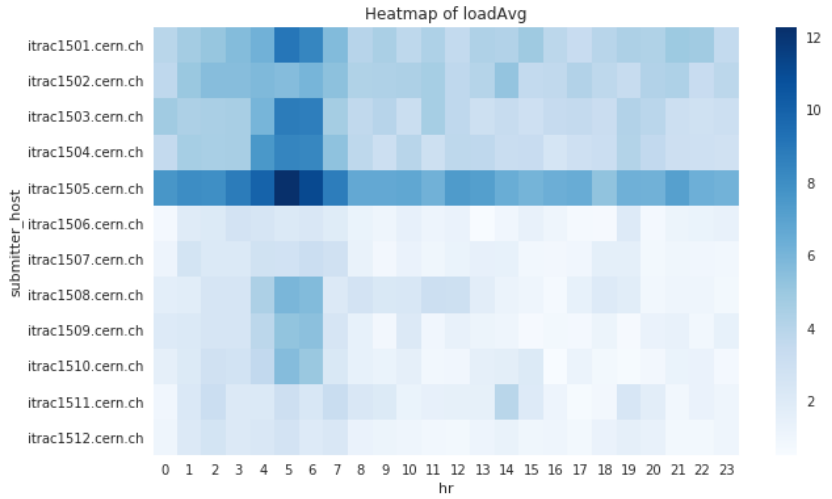
Apache Spark: 90 EXECUTORS 180 CORES Jobs: 1 COMPLETED

Job ID	Job Name	Status	Stages	Tasks	Submission Time	Duratio
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

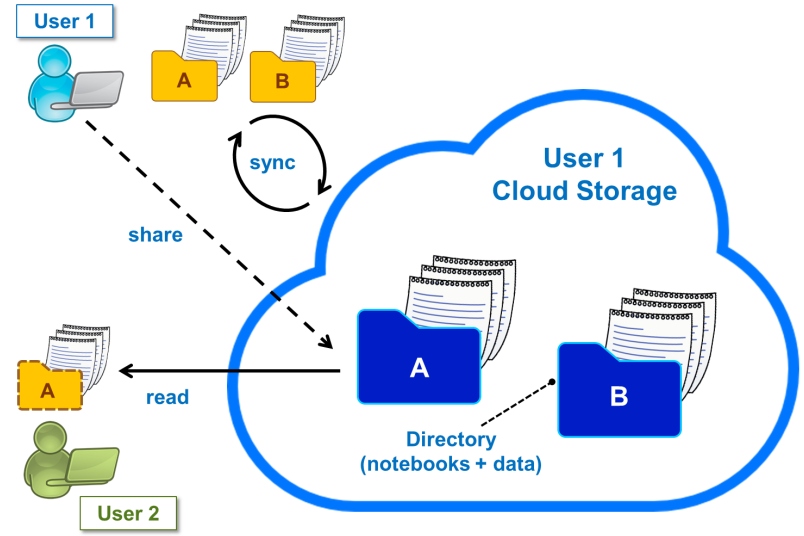
Visualizations

Cloud storage as your Home

> CERNBox is SWAN's home directory
- Storage for your notebooks and data
- 16k users and 6PB of user data

> 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



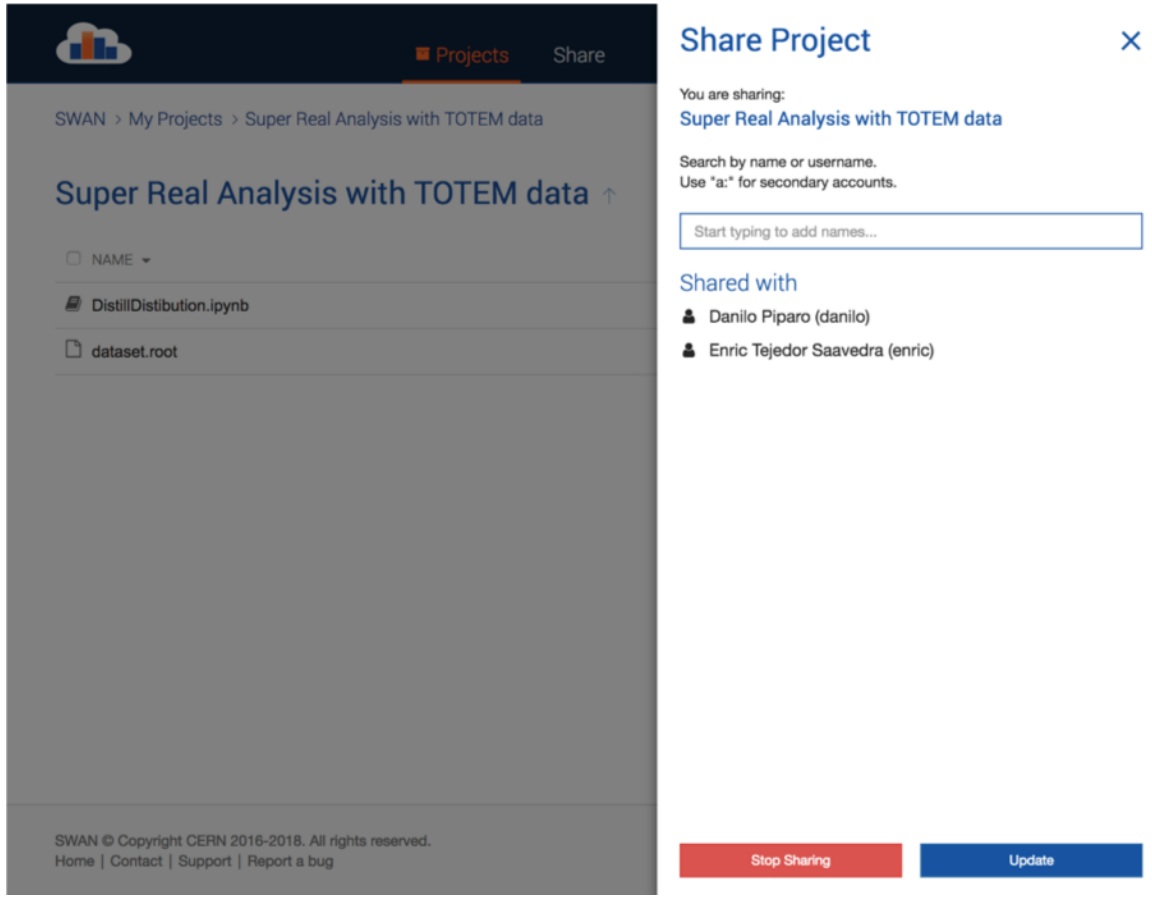
EOS CERNBox

CERN

Sharing made easy

> Sharing from inside SWAN interface
  - Integration with CERNBox
  - List shares from other users

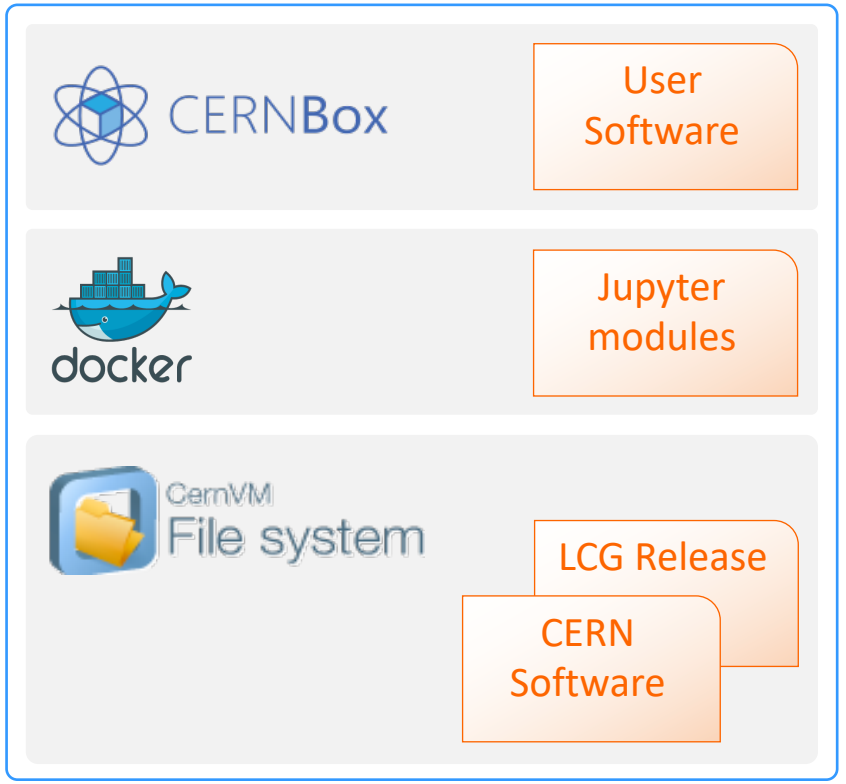
> Users can share “Projects”
  - Special kind of folder that contains notebooks and other files, like input data
  - Self contained



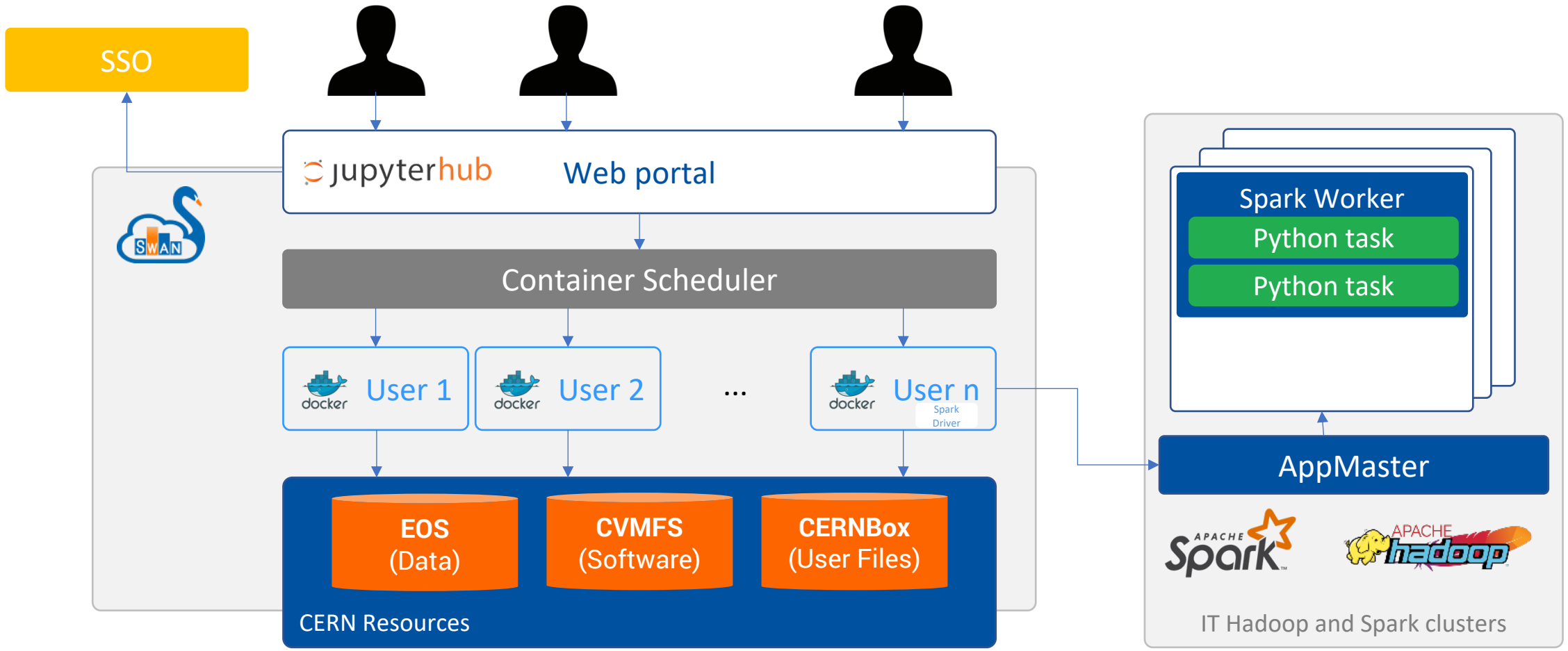
Software

> Software distributed through CVMFS
- Distributed read-only filesystem
- ”LCG Releases” - pack a series of compatible packages
- Reduced Docker Images size
- Lazy fetching of software

> Possibility to install libraries in user cloud storage
- Good way to use custom/not mainstream packages
- Configurable environment



Architecture

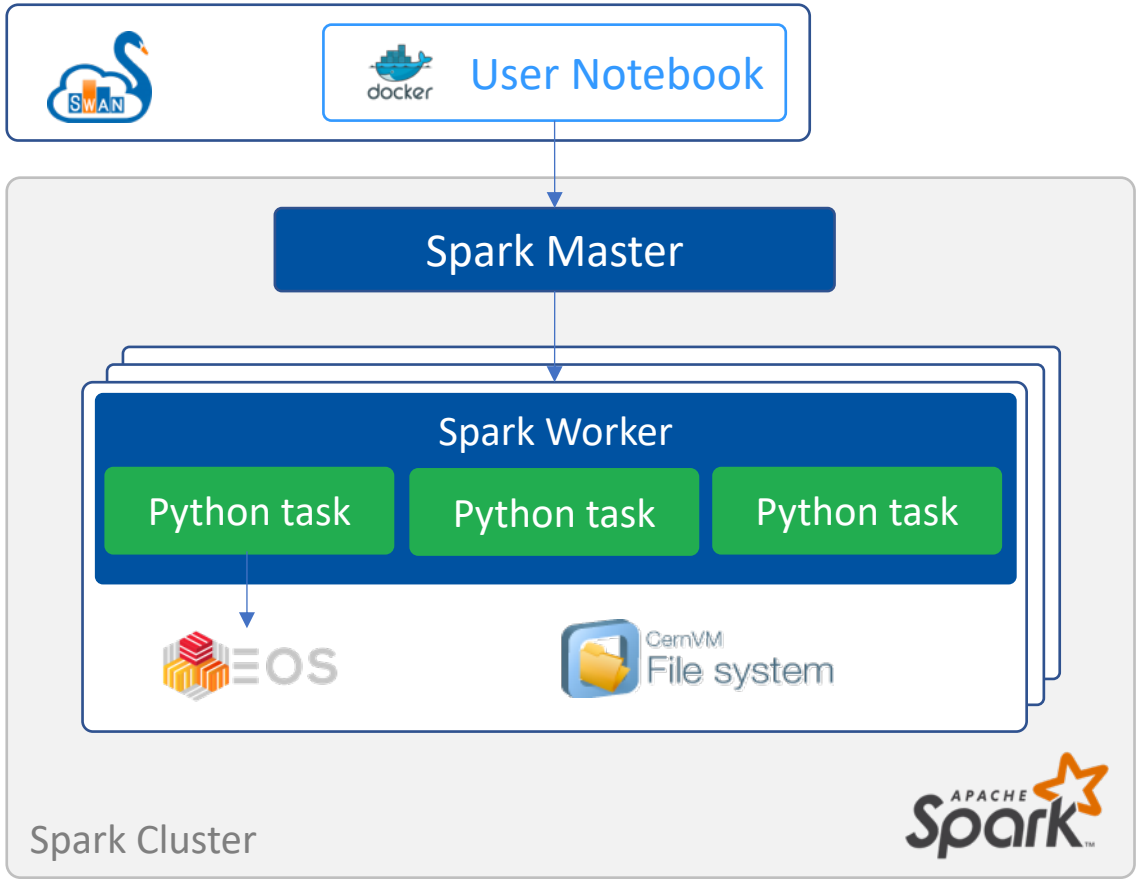


* User sessions are terminated after 6h of no activity to optimize usage of compute resources

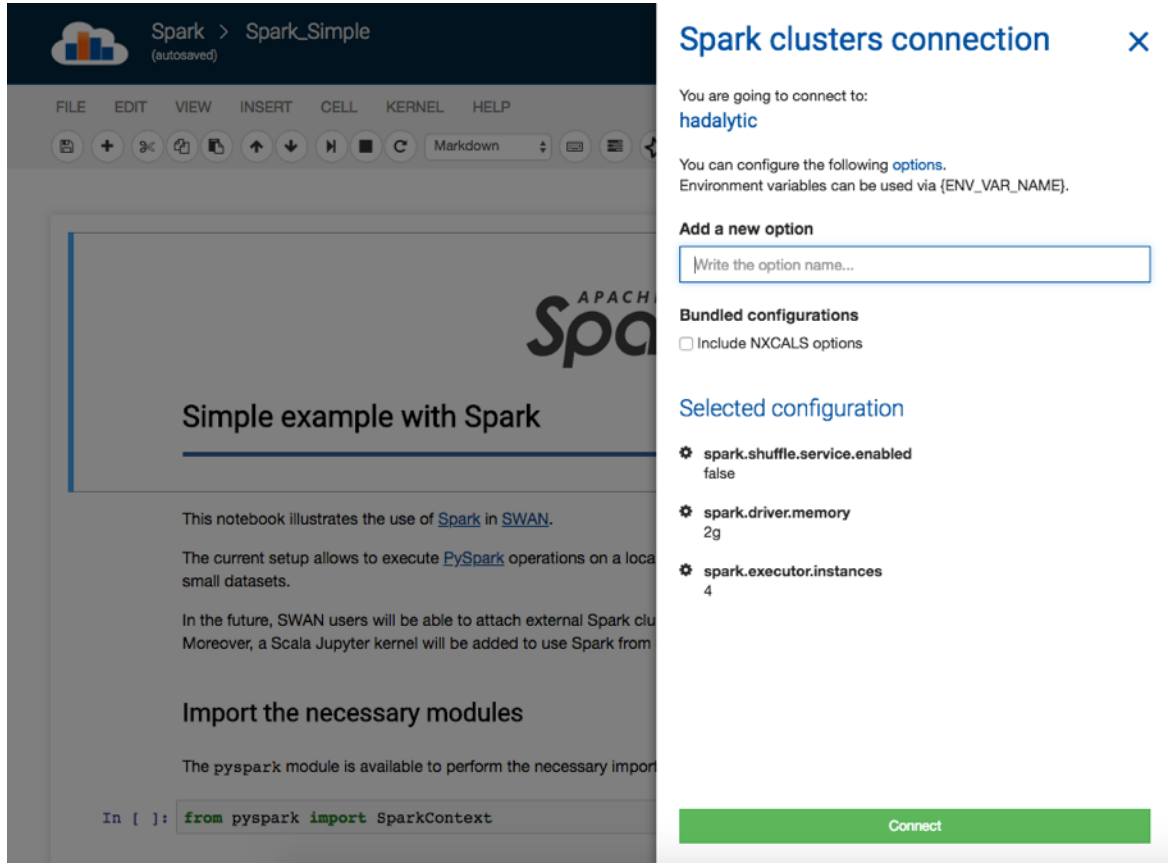
Access to Computing Resources

Integration with Spark

- Connection to CERN Spark Clusters
  - Spark: general purpose distributed computing framework
- Same environment across platforms (local/remote)
  - Software - CVMFS
- Graphical Jupyter extensions developed
  - Spark Connector
  - Spark Monitor
- Spark Clusters
  - NXCals: – Dedicated cluster for accelerator logging
  - Analytix: – General purpose YARN cluster
  - Cloud Containers: – General purpose Kubernetes cluster



Spark Connector



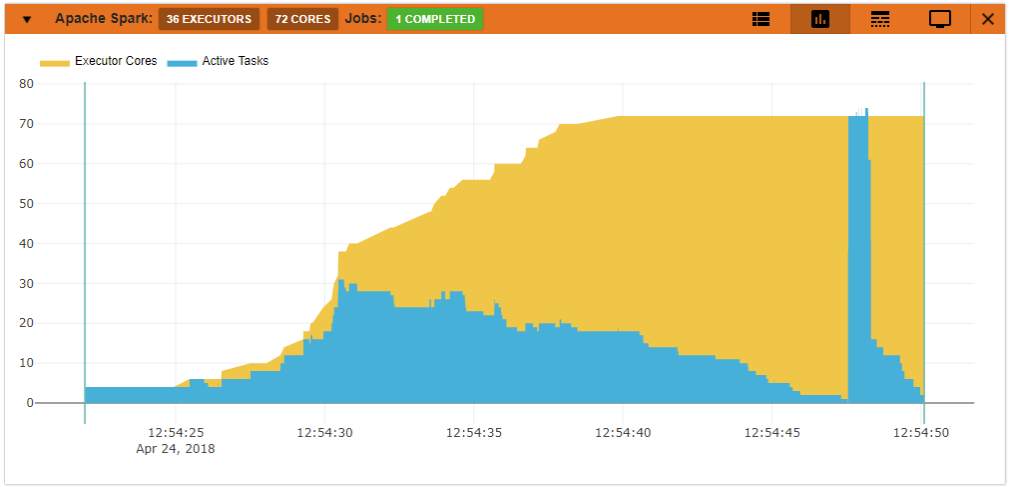
> Spark Connector – handling the spark configuration complexity

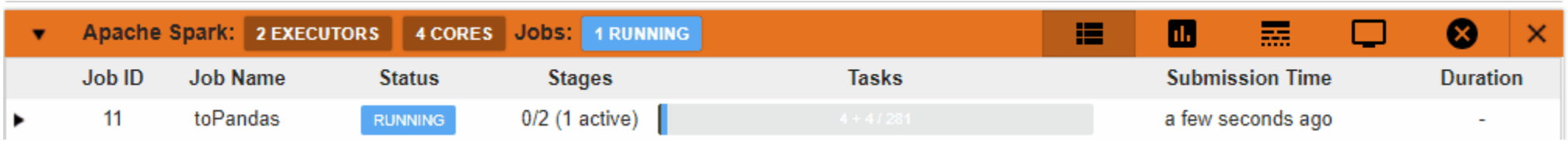
- User is presented with Spark Session (Spark) and Spark Context (sc)
- Ability to bundle configurations specific to user communities
- Ability to specify additional configuration

Spark Monitor

> Spark Monitor – jupyter notebook extension

- For live monitoring of spark jobs spawned from the notebook
- Access to Spark WEB UI from the notebook
- Several other features to debug and troubleshoot Spark application



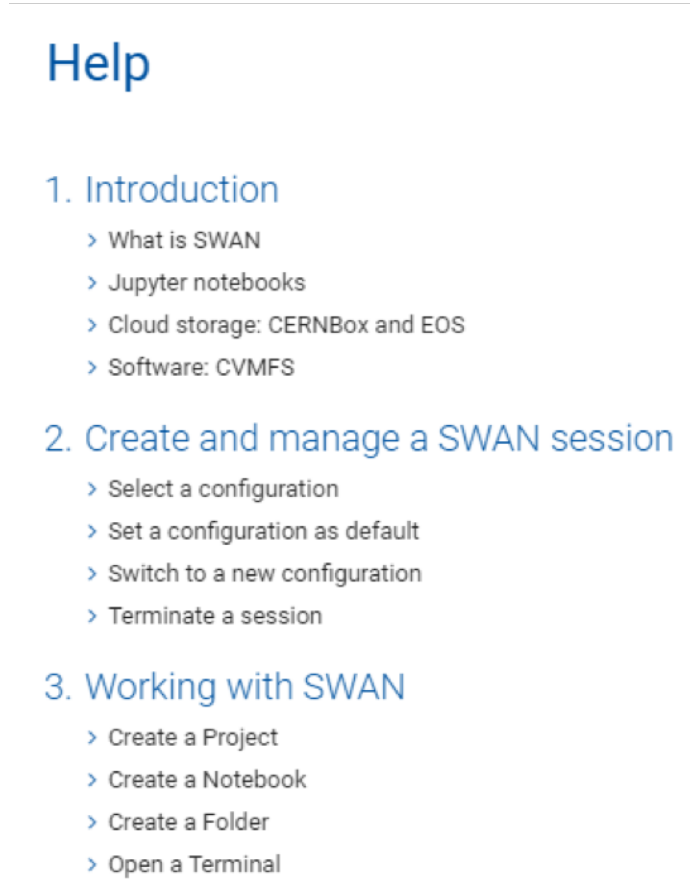


How to get help?

> SWAN Community
- https://cern.ch/swan-community
- Find solution to the commonly encountered issues / questions on the usage of Jupyter notebooks, LCG releases, storage and spark
- Report improvements / new features to the service
- E.g: How to install custom user packages

> Service Now
- Report issues to the service
- E.g: Unable to start a session

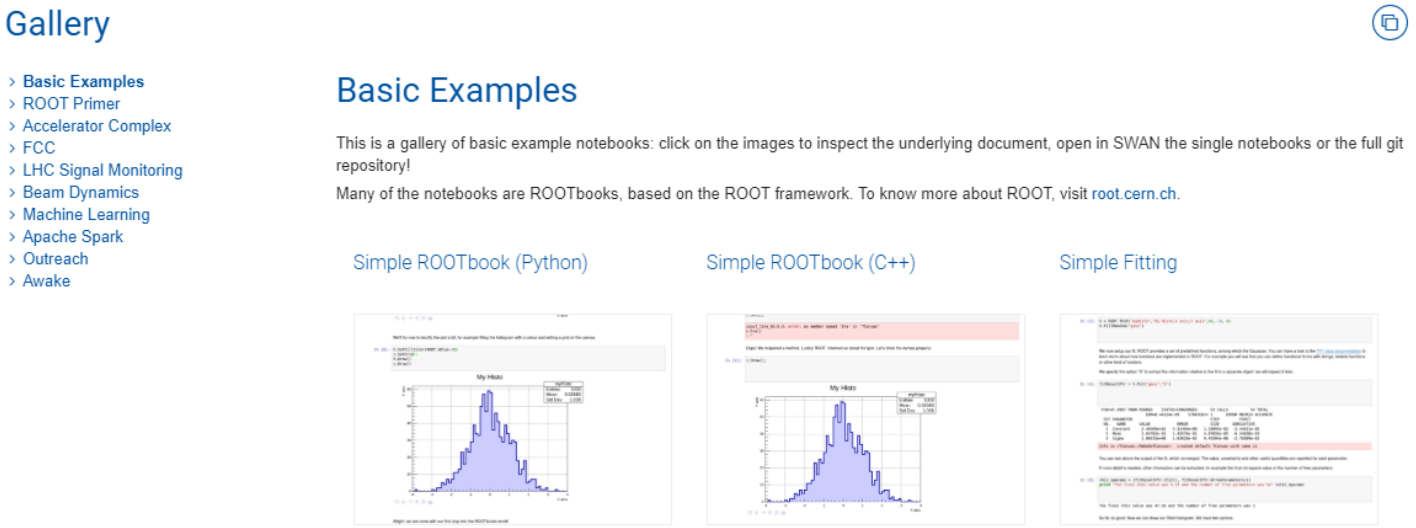
> Help on various functionality of the tool ?



How to get started?

> Gallery of sample notebooks for varied usage of SWAN

- Quick way to be productive
- Also accessible from cern.ch/swan



Where to find us

Where to find us

> Contacts
- swan-admins@cern.ch
- http://cern.ch/swan

> Repository
- https://gitlab.cern.ch/swan

> Science Box
- https://cern.ch/sciencebox

CERN

SWAN and its analysis ecosystem

Thank you

Prasanth Kothuri
prasanth.kothuri@cern.ch