

SWAN interactive data analysis on the web



Diogo Castro
On behalf of the SWAN team

<https://cern.ch/swan>

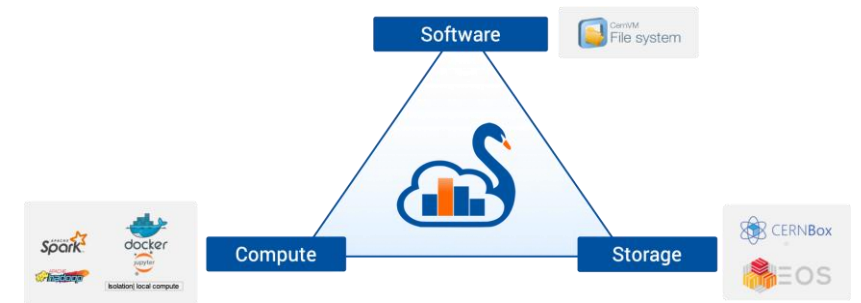
Oct 14th, 2019
ESCAPE meeting





SWAN in a Nutshell

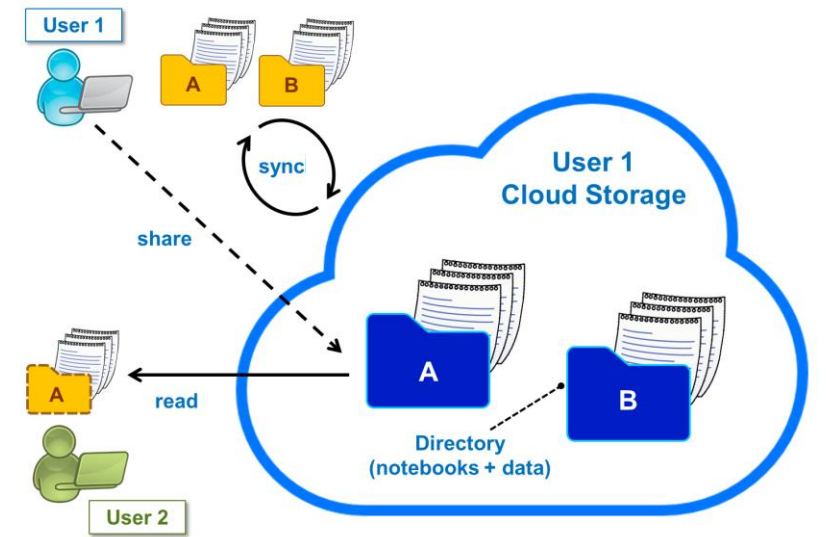
- › CERN's Jupyter Notebook service
 - Used for final steps of an Analysis, Exploration, Teaching, Documentation and Reproducibility
 - Easy sharing of scientific results: plots, data, code
- › Support for multiple analysis ecosystems and languages
 - Python, ROOT C++, R and Octave
- › Integration with CERN resources
 - Software, storage, mass processing power





Cloud storage as your Home

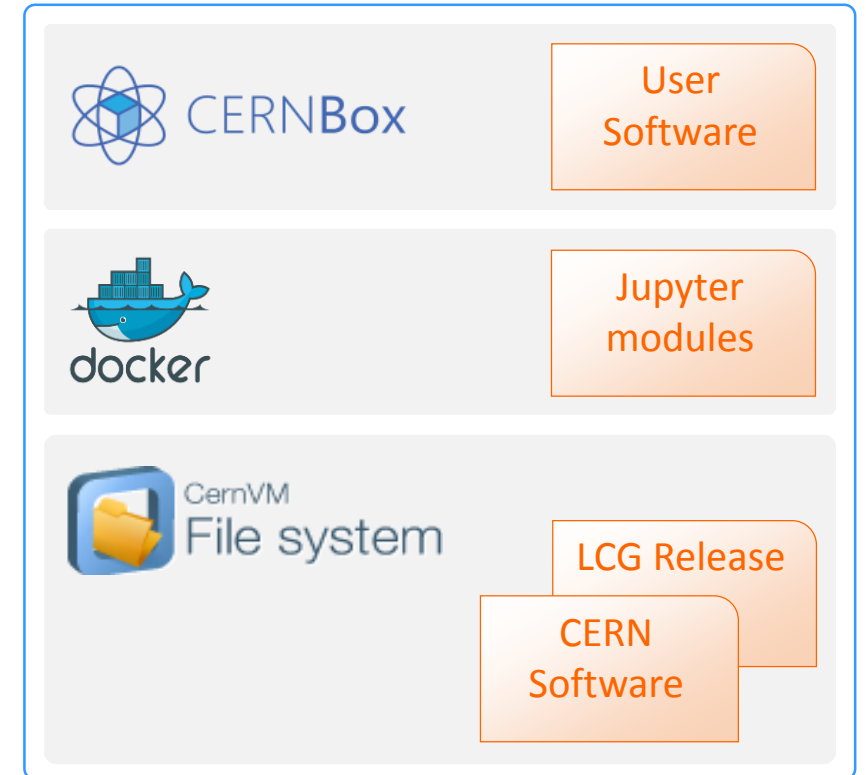
- > CERNBox is SWAN's home directory
 - Based on EOS disk storage system
- > Sync & Share
 - Files synced across devices and the Cloud
 - Collaborative analysis
- > Sharing integration within SWAN UI
 - Users can share “Projects”(special kind of folder that contains notebooks and other files, like input 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





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

The screenshot shows a Jupyter notebook titled "Spark > Spark_Simple (autosaved)". The notebook content includes a section "Simple example with Spark" and text explaining that the setup allows for executing PySpark operations on local small datasets. A sidebar on the right, titled "Spark clusters connection", shows the configuration for connecting to a cluster named "hadalytic". It lists bundled configurations and a selected configuration with the following settings:

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

A "Connect" button is visible at the bottom of the sidebar.

The screenshot shows a Jupyter notebook cell with the following 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()")
```

Below the code, a status bar indicates "Apache Spark: 90 EXECUTORS 180 CORES Jobs: 1 COMPLETED". A table shows the execution details:

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





Service evolution

- > Jupyterlab
 - Next-generation interface for Project Jupyter
 - Concurrent editing
- > NVidia GPU Support
 - Already integrated with ScienceBox
 - New LCG stack with CUDA enabled machine learning software
- > Batch jobs submission
- > Configurable software environment for Projects
 - Associated with Conda Environments
 - Easy installation and sharing



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>

SWAN interactive data analysis on the web

Thank you

Diogo Castro
diogo.castro@cern.ch