

# SWAN

## Data Analysis, in the Cloud

<http://swan.web.cern.ch>

D. Piparo for the SWAN Team  
HEPData Workshop



- **S**ervice for **W**eb based **A**nalysis
- Provides a **cloud based analysis model**
- **Data, software and computing resources in the cloud**
- **Only a web browser is needed:**
  - **Jupyter notebooks** and ordinary linux shells provided
- Started as ROOT-as-a-Service in 2015
- A joint effort of EP-SFT and IT-ST: D.P., E. Tejedor, M. Lamanna, P. Mato, J.T. Moscicki, L. Mascetti.

A web-based **interactive computing interface and platform** that **combines code, equations, text and visualisations**.



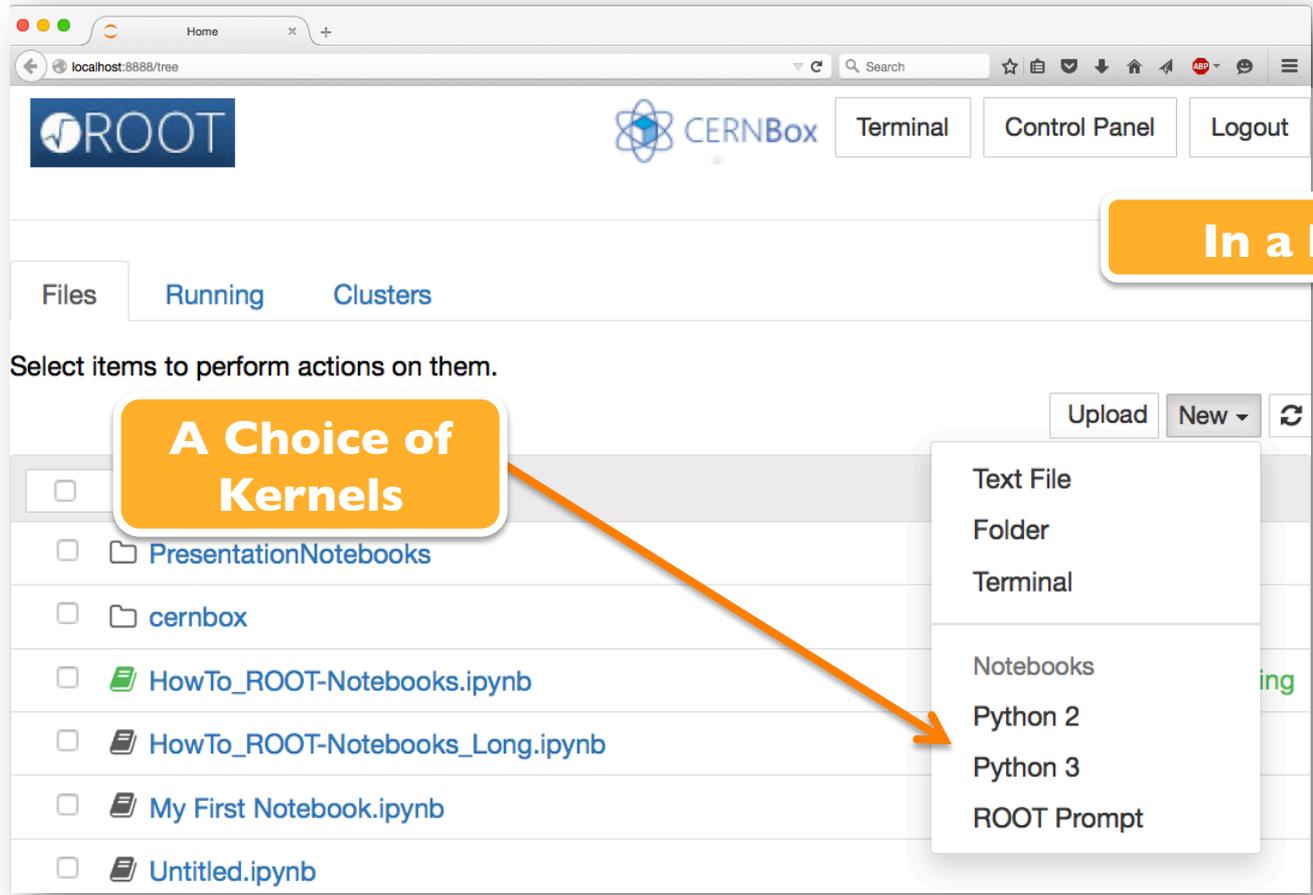
<http://www.jupyter.org>

Many supported languages: Python, Haskell, Julia, R ... One generally speaks about a “kernel” for a specific language

In a nutshell: an “interactive shell opened within the browser”

Also called:

“Jupyter Notebook” or “IPython Notebook”



In a browser

A Choice of Kernels

- Text File
- Folder
- Terminal
- Notebooks
- Python 2
- Python 3
- ROOT Prompt

*Kernels* are processes that run interactive code in a particular programming language and return output to the user. Kernels also respond to tab completion and introspection requests.

The image shows a Jupyter Notebook interface in a web browser. The browser address bar shows 'localhost:8888/tree'. The notebook title is 'Notebook Functionalities'. The interface includes a menu bar with 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', and 'Help'. The 'Kernel' menu is open, showing 'Python 2' selected and circled in red. Below the menu bar is a toolbar with icons for saving, adding, deleting, and running cells. The main content area displays a welcome message and a code cell.

**ROOT** Notebook Functionalities Control Panel Logout

File Edit View Insert Cell Kernel Help **Python 2**

Code Cell Toolbar: None

**Text and Formulas**

## Welcome to the Notebook Technology

This is a markdown cell. You can add LaTeX code:  $\sum_{n=-\infty}^{\infty} |x(n)|^2$

```
In [1]: def thisFunction():
        return 42
```

```
In [2]: thisFunction()
```

```
Out[2]: 42
```

**Code**

```
In [1]: def thisFunction():  
        return 42
```

```
In [2]: thisFunction()
```

```
Out[2]: 42
```

```
In [3]: %%bash  
        curl rootaasdemo.web.cern.ch/rootaasdemo/SaaSFee.jpg \  
> SF.jpg
```

```
% Total      % Received % Xferd  Average Speed   Time  
Time        Time      Current           Dload  Upload   Total  
100 128k  100  128k    0      0  2731k      0  --:--:--  
--:--:-- --:--:-- 2787k
```

Shell Commands

```
In [4]: from IPython.display import Image  
        Image(filename="./SF.jpg",width=225)
```

```
Out[4]:
```



Images

```
In [1]: def thisFunction():  
        return 42
```

```
In [2]: thisFunction()
```

```
Out[2]: 42
```

```
In [3]: %%bash  
        curl rootaasdemo.web.cern.ch/rootaasdemo/SaaSFee.jpg \
```

**No excuses possible when it comes to describe all steps in an analysis!**

```
      Spent   Left  Speed  
100 128k 100 128k   0    0 2731k    0 --:--:--  
--:--:-- --:--:-- 2787k
```

```
In [4]: from IPython.display import Image  
        Image(filename='./SF.jpg',width=225)
```

```
Out[4]:
```



- Platform independent: **only with a web browser**
  - Analyse data **via the Notebook web interface**
  - **No need to install and configure software**
- Calculations, input and results **“in the cloud”**
- Allow **easy sharing of scientific results**: plots, data, code
  - Storage is crucial, both mass (EOS) and synchronised (CERNBox)
- **Simplify teaching** of data processing and programming
- **Eases analysis reproducibility**
- **C++, Python** and other languages or analysis **“ecosystems”**
  - Interfaced to widely adopted scientific libraries
  - e.g. **Rivet, MC Generators, Pandas, Numpy, ROOT, matplotlib, ...**





# Novel Idea, Technologies in Production

SWAN relies on production technologies at CERN. In a nutshell:

- CPU: CERN OpenStack instance 
- Software: LCG releases on CVMFS
  - ~200 packages: fastjet, ROOT, Geant4, Pythia, Rivet, matplotlib...
- Storage, in two flavours:
  - Mass: EOS (the storage backend of HEPData too!) 
  - Synchronised: CERNBox (the “home directory” in SWAN)



- Same storage backend of **HEPData**.
- **Software building blocks** for data analysis
- **Share** code, results and derived data with colleagues



# The Architecture

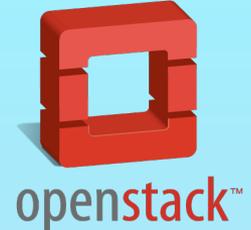
**CERN Auth**



**CERN Cloud**

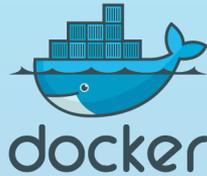
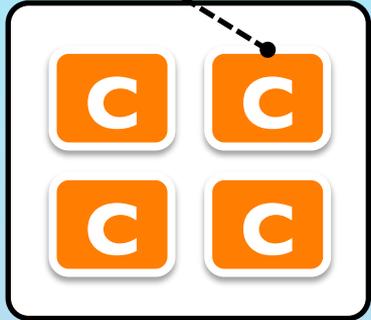
Jupyter

**Web Portal** Jupyterhub



**Notebook Container**

**Container Scheduler**



EOS (Data) CVMFS (Software)

CERNBox (User Files) C C C

SWAN and HEPData: “Proof of Concept” analysis

- SWAN provides a cloud based analysis model
  - Unique requirement to access it: a web browser
- Experiments data, users' data, HEPData accessible on EOS
- Rich collection of software packages distributed via CVMFS
- Synchronisation and sharing possible thanks to CERNBox

Promising interplay between HEPData and SWAN!