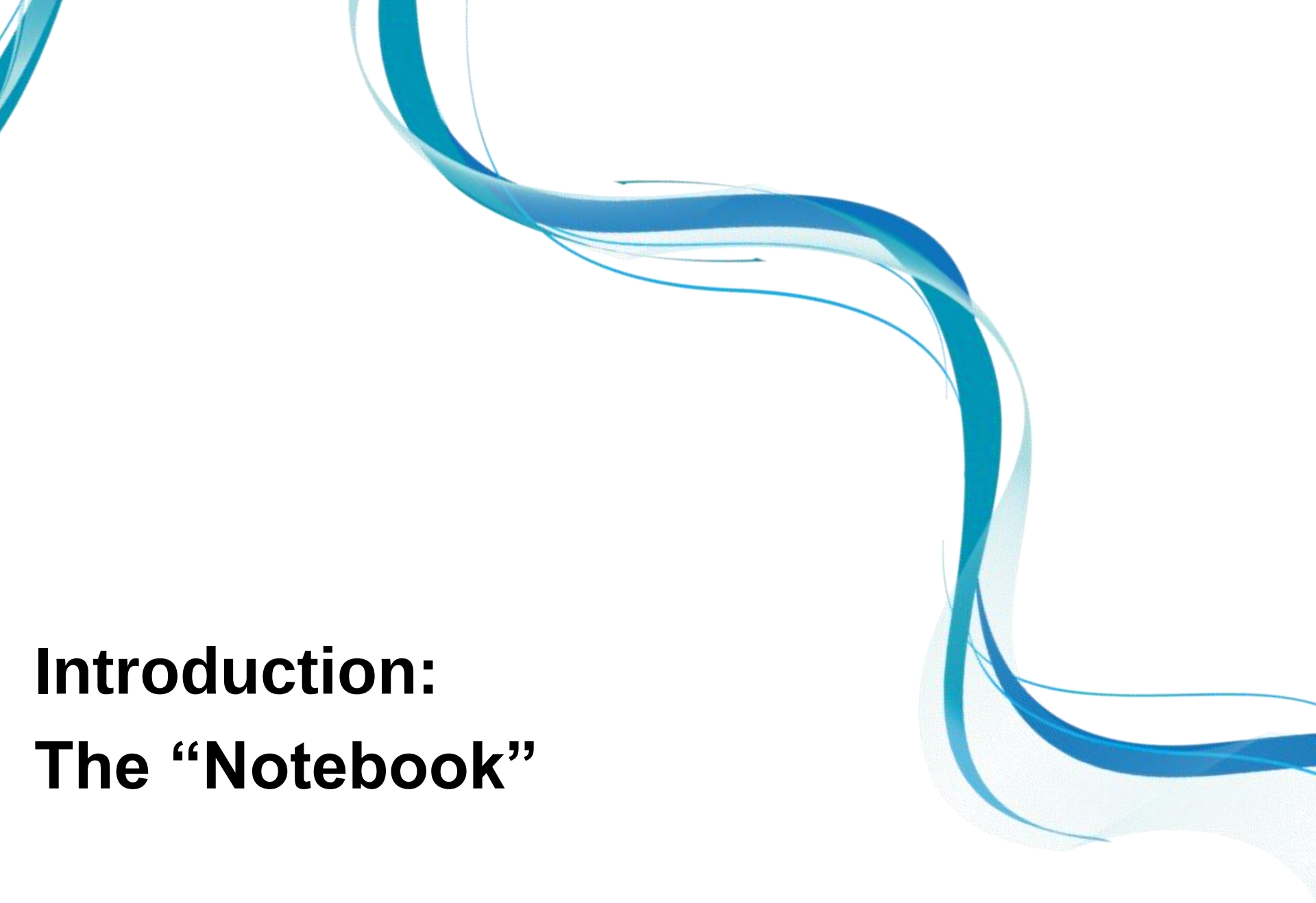
ROOTaaS
ROOT as a Service

E. Tejedor, D. Piparo, P. Mató for the ROOT Team

PH-SFT meeting

12/10/2015



Introduction:
The “Notebook”

The Notebook

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

Many supported languages: Python, Haskell, Julia… 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”

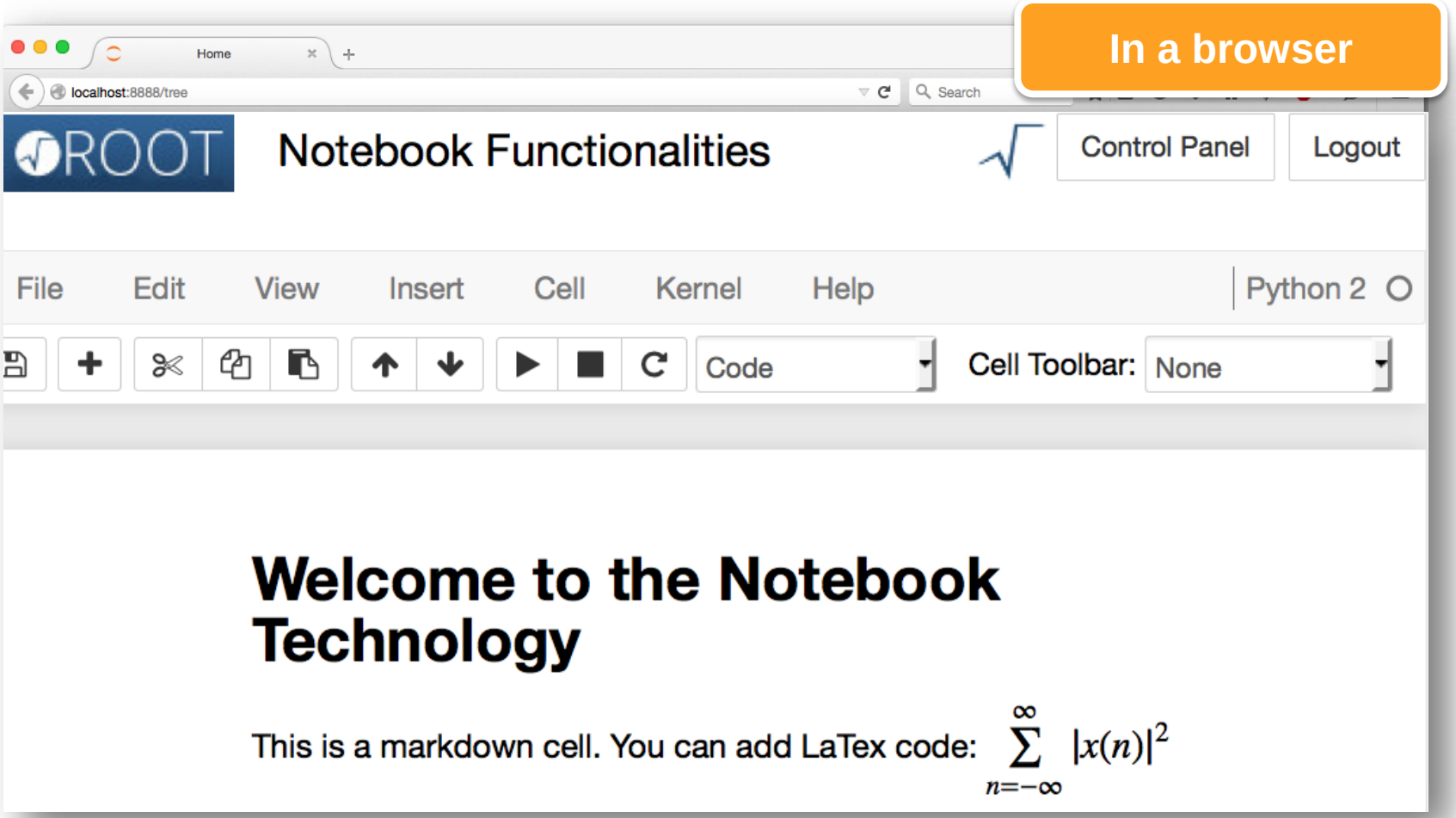
http://www.jupyter.org

Start a Notebook in a Laptop

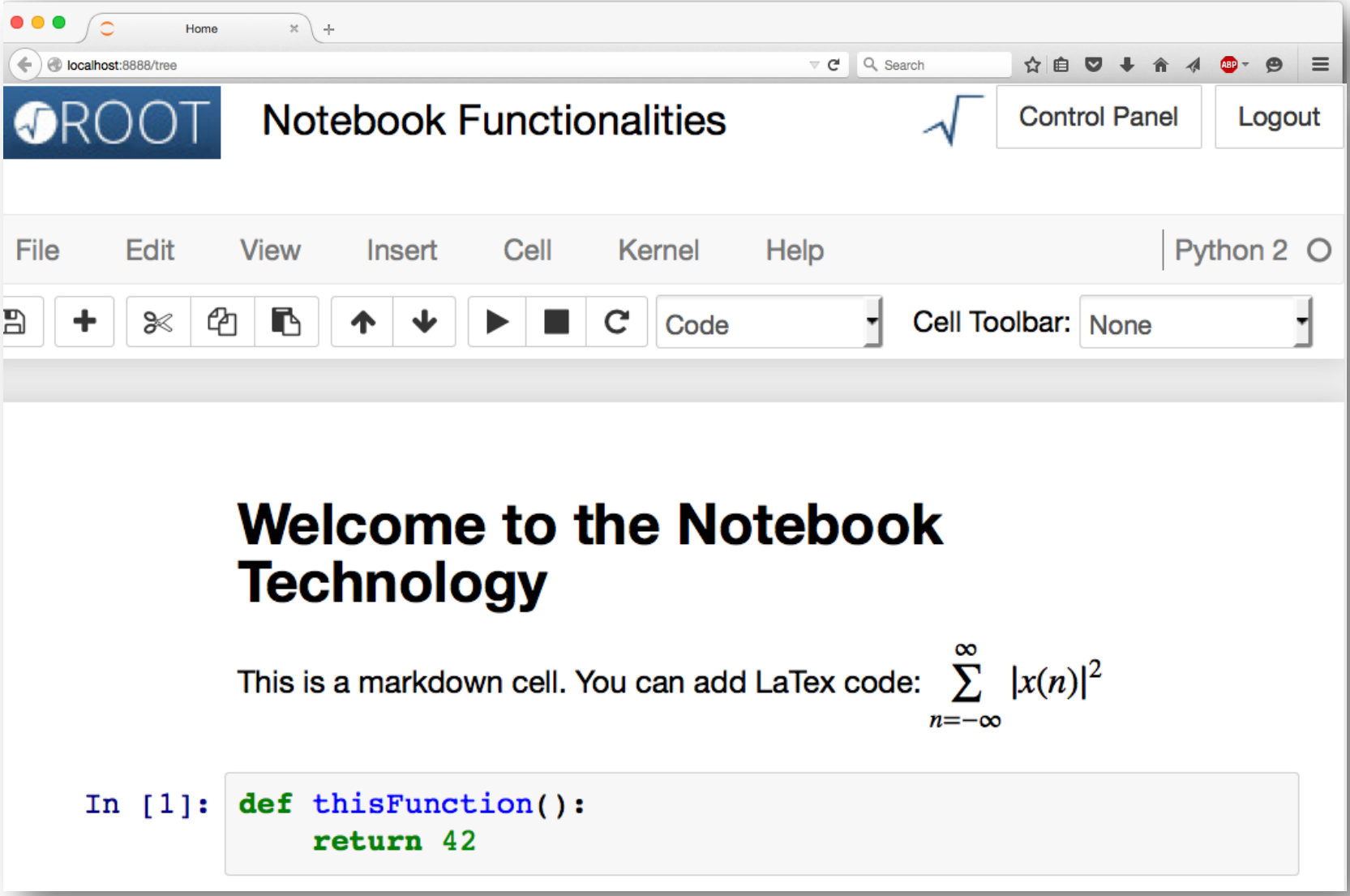
```
$ ipython notebook
```

This command:

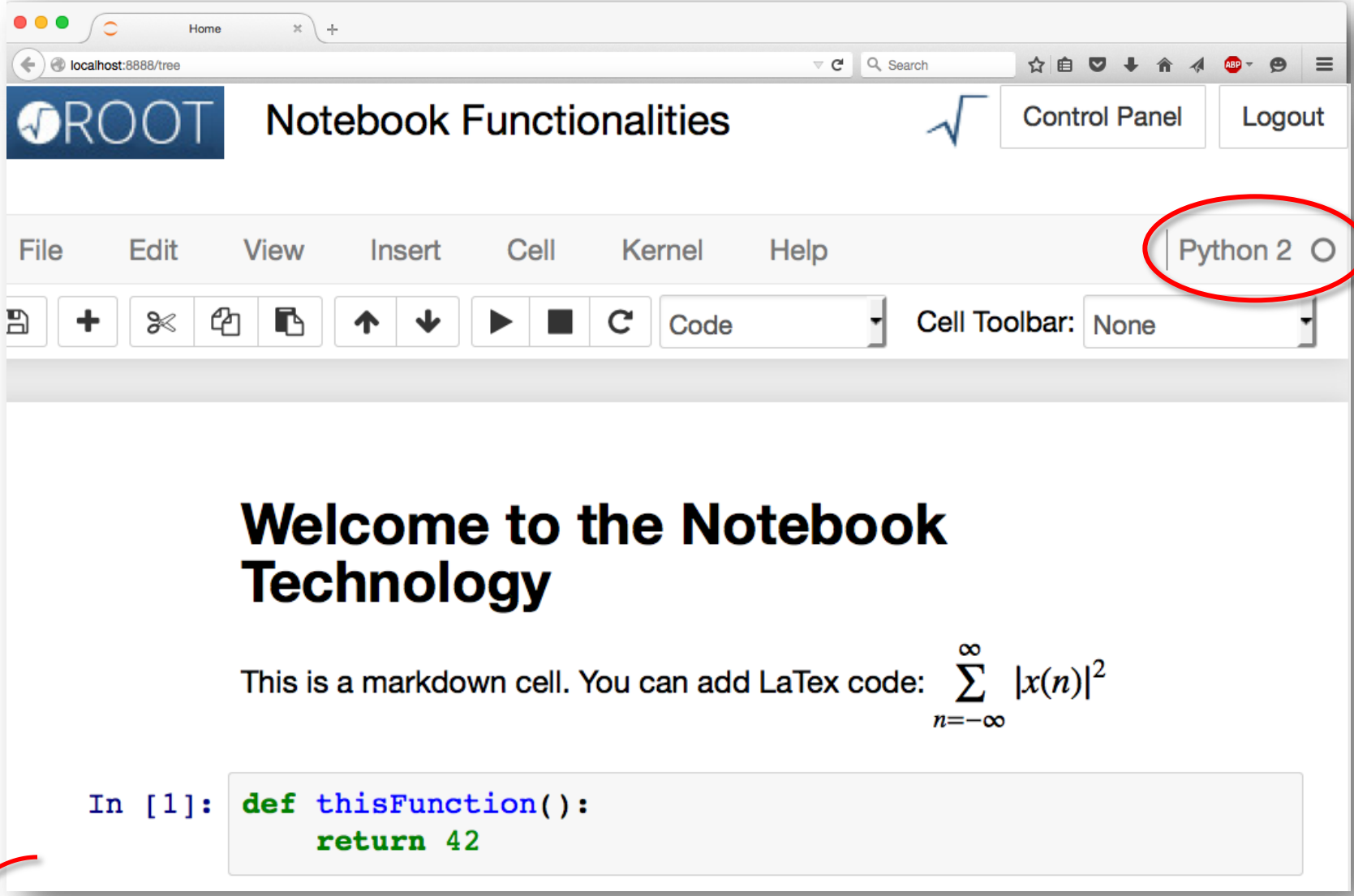
1. Starts a local notebook server
2. Connects to it via the browser



Text and Formulas

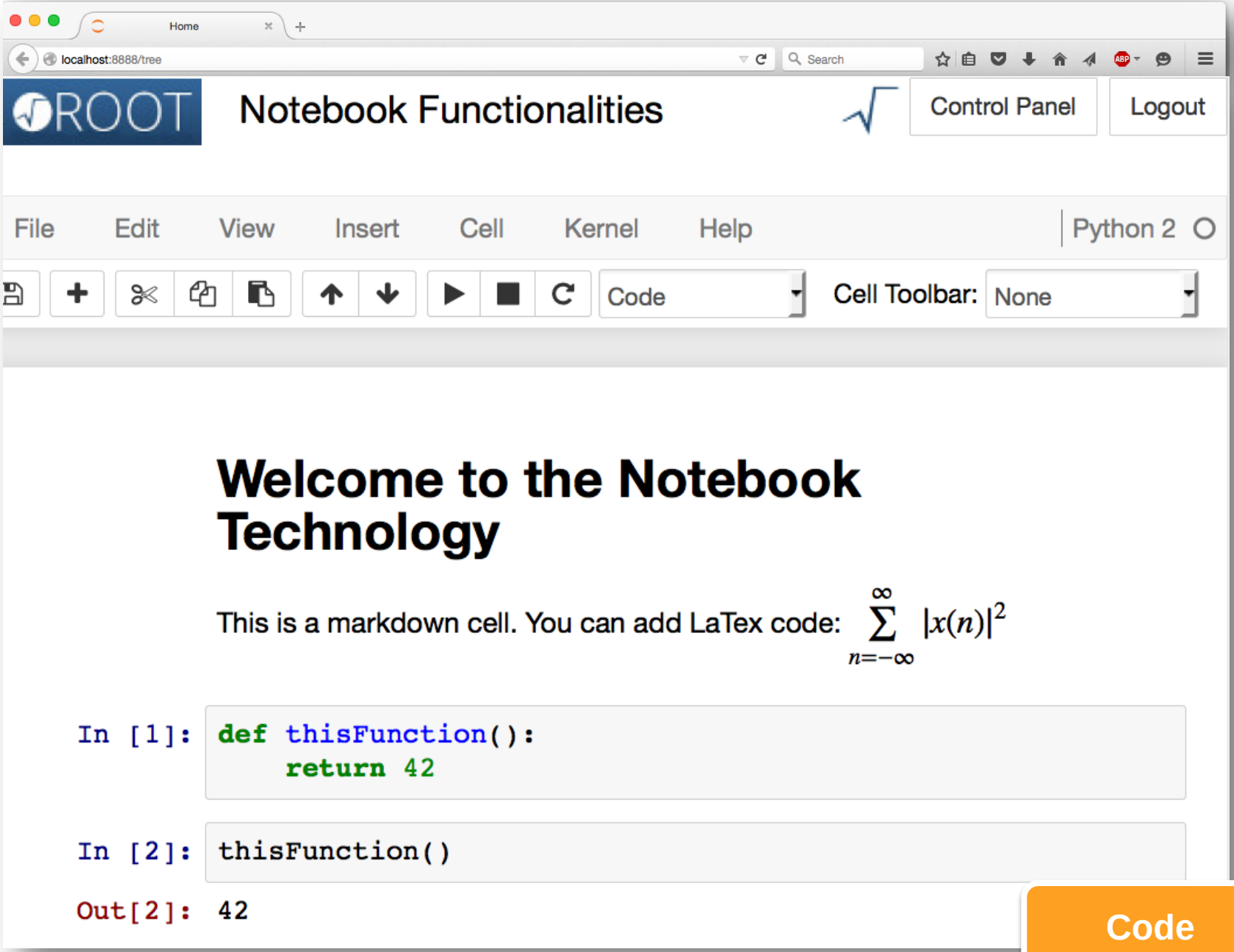

Home
localhost:8888/tree
Search
ROOT
Notebook Functionalities
Control Panel
Logout
File
Edit
View
Insert
Cell
Kernel
Help
Python 2
Code
Cell Toolbar:
None
Welcome to the Notebook Technology
This is a markdown cell. You can add LaTex code: $\sum_{n=-\infty}^{\infty} |x(n)|^2$
In [1]:
def thisFunction():
return 42

Code



This is a notebook in Python

Code


Home
localhost:8888/tree
Search
ROOT
Notebook Functionalities
Control Panel
Logout
File
Edit
View
Insert
Cell
Kernel
Help
Python 2
Code
Cell Toolbar:
None
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
```

We can invoke commands in the shell…

Shell Commands

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

… and get their output

Shell Commands

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
  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)
```

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
  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]:

Images

In a browser

Text and Formulas

Code

Shell Commands

Images

In [1]:

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

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

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

```
  % Total    % Rece                ed   Time
  Time     Time  Cu
                                 Dload  Upload   Total
  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)
```

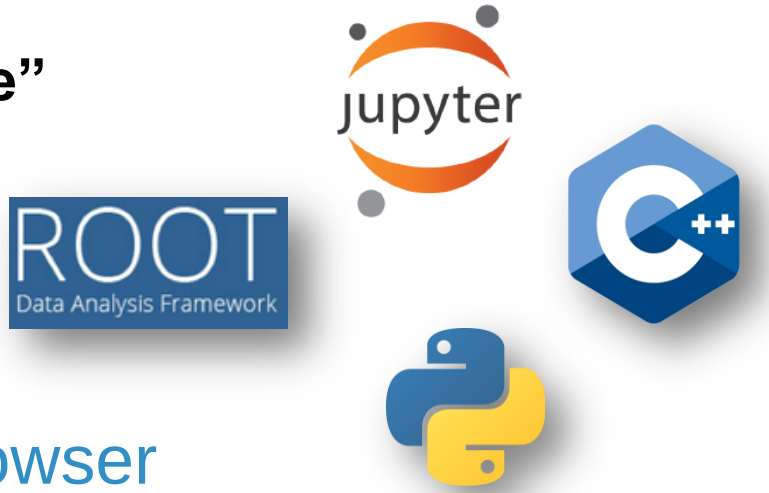
Out[4]:

This Talk

- The ROOTaaS project and why it is needed
- Integration of ROOT with the Notebook technology
  - Programming model and usability for data analysis
- ROOTaaS within the CERN IT services' portfolio
  - Spotlight on storage

The ROOTaaS Project

Data mining with ROOT "as a service"



Interface: Notebooks

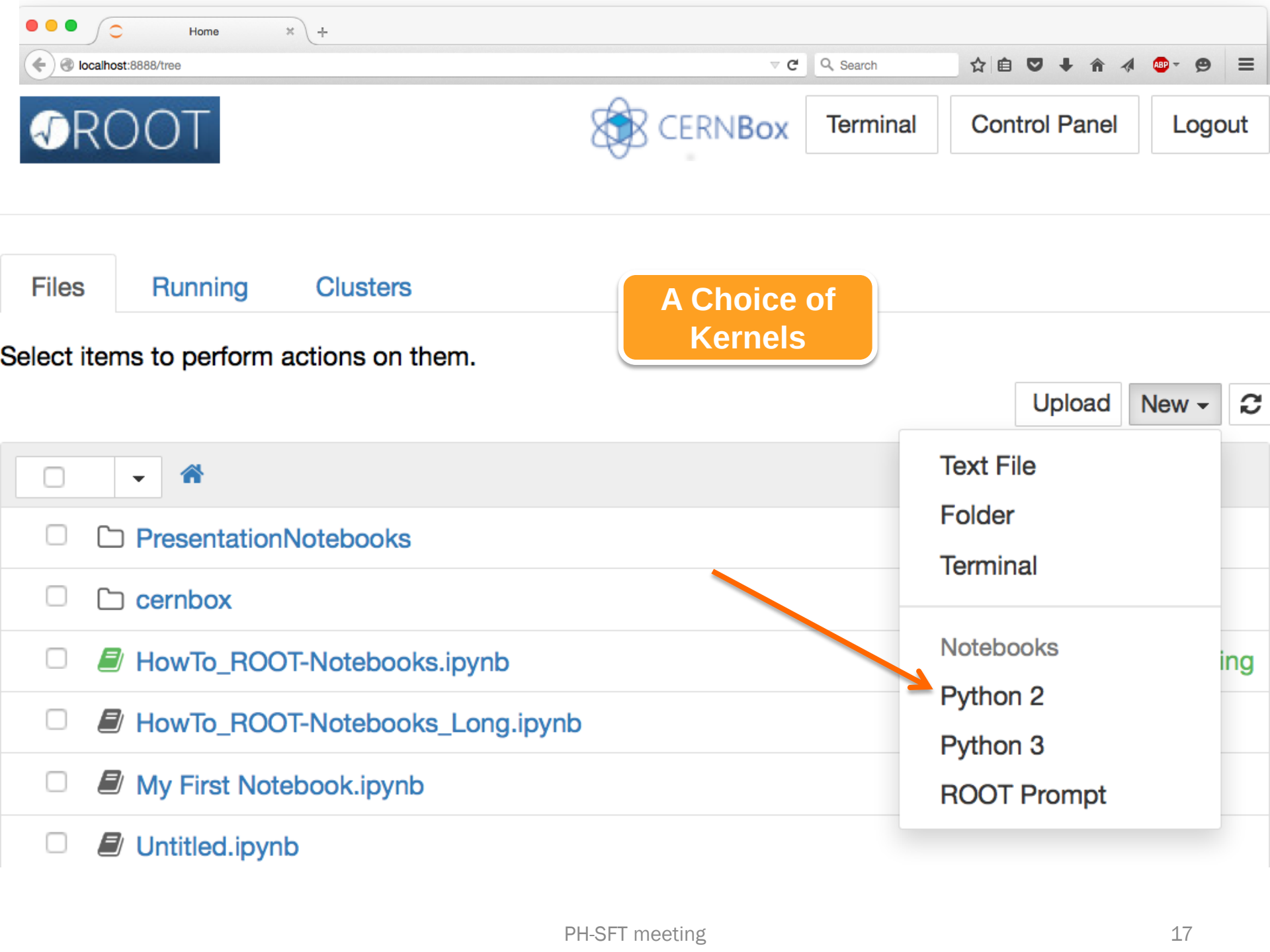
Goals:

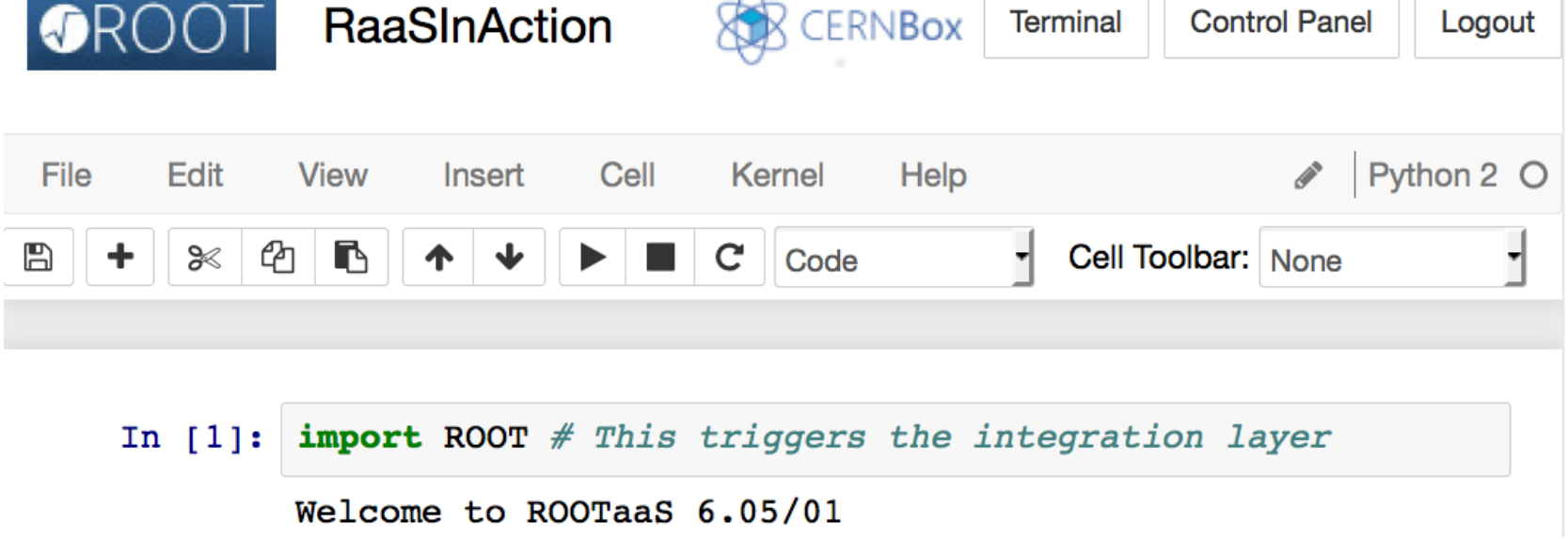
- Use ROOT only with a web browser
  - Platform independent ROOT based data analysis
  - Calculations, input and results "in the cloud"
- Allow easy sharing of scientific results: plots, data, code
  - Storage is crucial
- Simplify teaching of data processing and programming
- C++, Python and other languages interfaced to ROOT

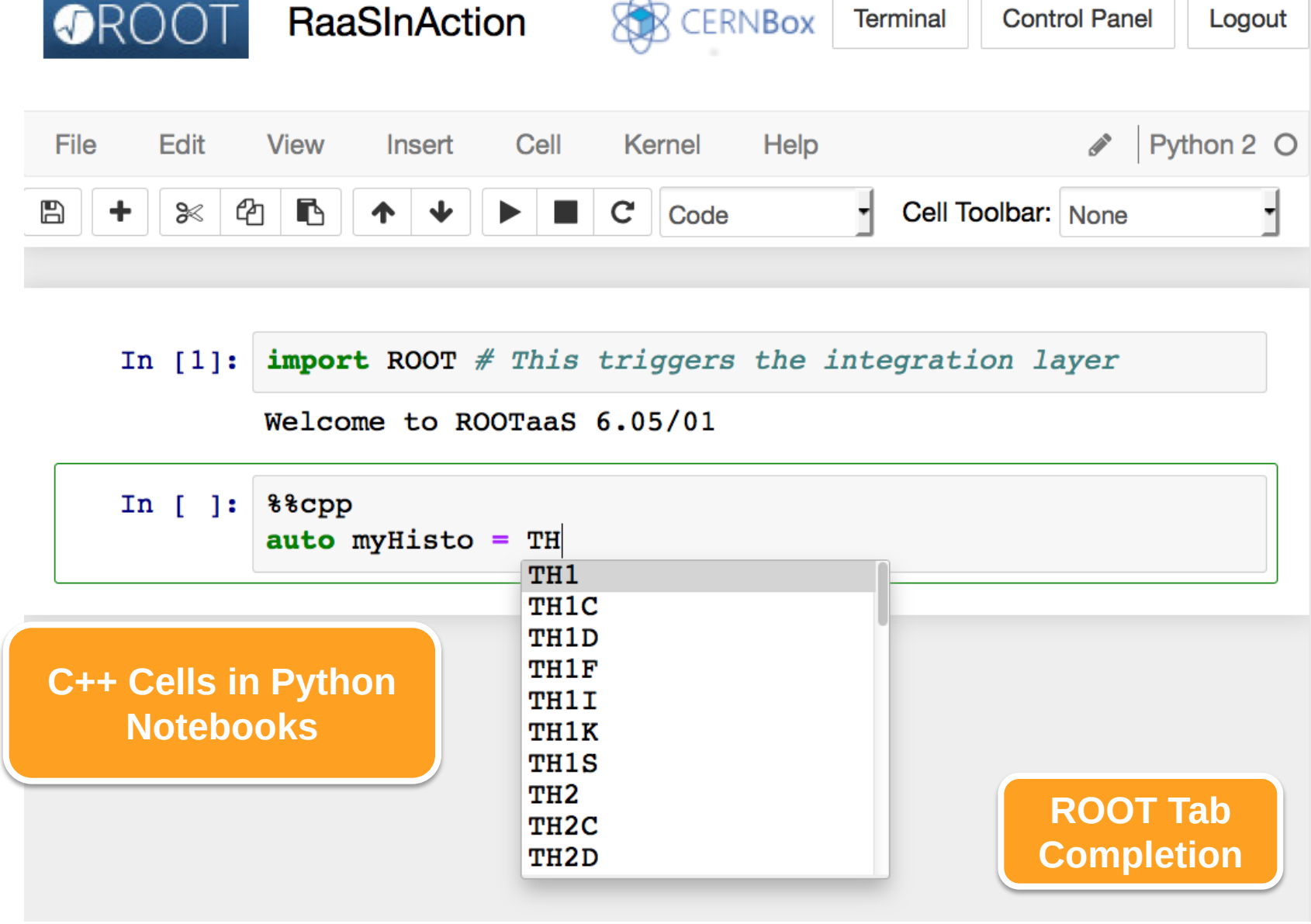
Integration of ROOT with Notebooks

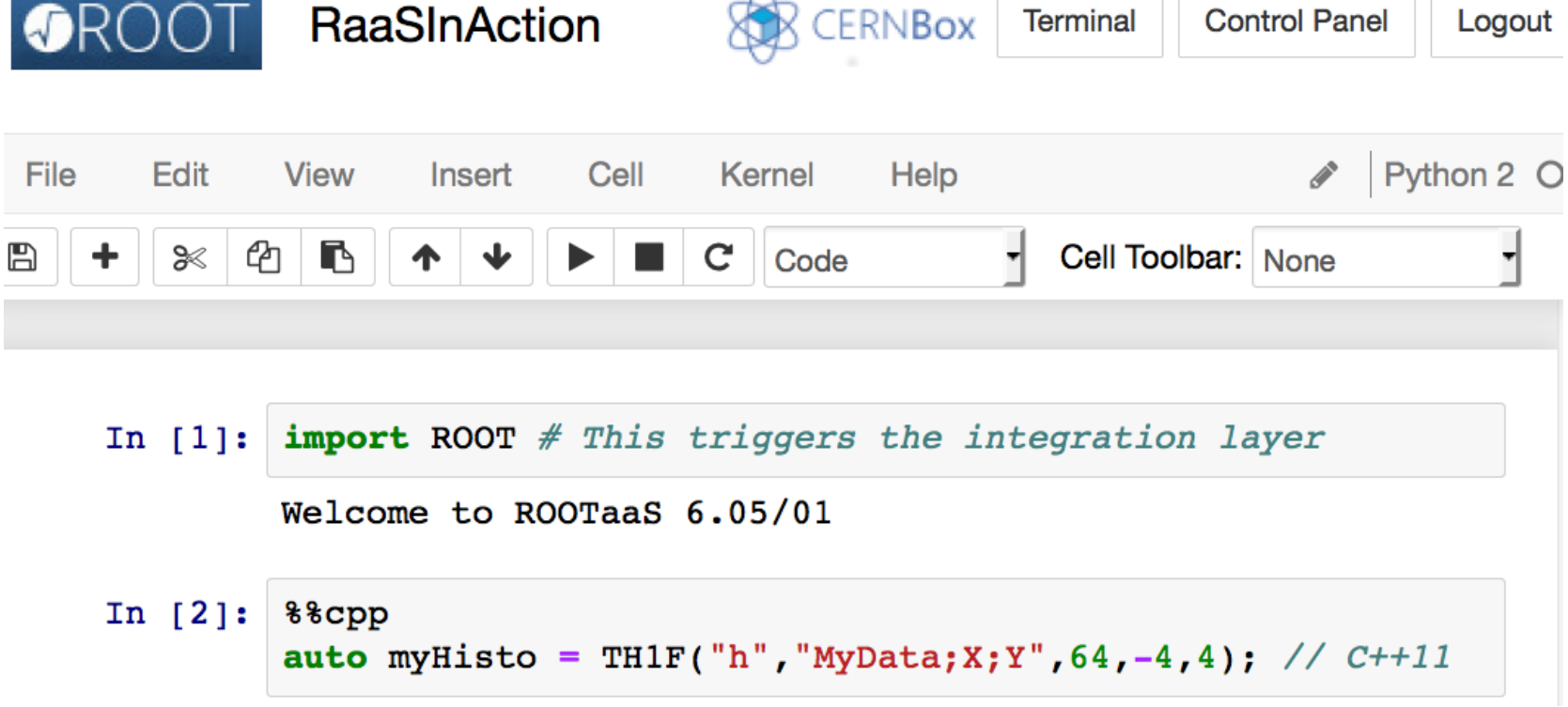
ROOT

iPyROOT
(ROOT-Notebooks integration)


Home
localhost:8888/tree
Search
ROOT
CERNBox
Terminal
Control Panel
Logout
Files
Running
Clusters
A Choice of Kernels
Select items to perform actions on them.
Upload
New
Text File
Folder
Terminal
Notebooks
Python 2
Python 3
ROOT Prompt
PresentationNotebooks
cernbox
HowTo_ROOT-Notebooks.ipynb
HowTo_ROOT-Notebooks_Long.ipynb
My First Notebook.ipynb
Untitled.ipynb


ROOT
RaaSInAction
CERNBox
Terminal
Control Panel
Logout
File
Edit
View
Insert
Cell
Kernel
Help
Python 2
Code
Cell Toolbar:
None
In [1]:
import ROOT # This triggers the integration layer
Welcome to ROOTaaS 6.05/01


ROOT
RaaSInAction
CERNBox
Terminal
Control Panel
Logout
File
Edit
View
Insert
Cell
Kernel
Help
Python 2
Code
Cell Toolbar:
None
In [1]:
import ROOT # This triggers the integration layer
Welcome to ROOTaaS 6.05/01
In [ ]:
%%cpp
auto myHisto = TH
TH1
TH1C
TH1D
TH1F
TH1I
TH1K
TH1S
TH2
TH2C
TH2D
C++ Cells in Python Notebooks
ROOT Tab Completion


ROOT
RaaSInAction
CERNBox
Terminal
Control Panel
Logout
File
Edit
View
Insert
Cell
Kernel
Help
Python 2
Code
Cell Toolbar:
None
In [1]:
import ROOT # This triggers the integration layer
Welcome to ROOTaaS 6.05/01
In [2]:
%%cpp
auto myHisto = TH1F("h","MyData;X;Y",64,-4,4); // C++11

ROOT RaaSInAction CERNBox Terminal Control Panel Logout

File Edit View Insert Cell Kernel Help Python 2

Code Cell Toolbar: None

In [1]:
```
import ROOT # This triggers the integration layer
```

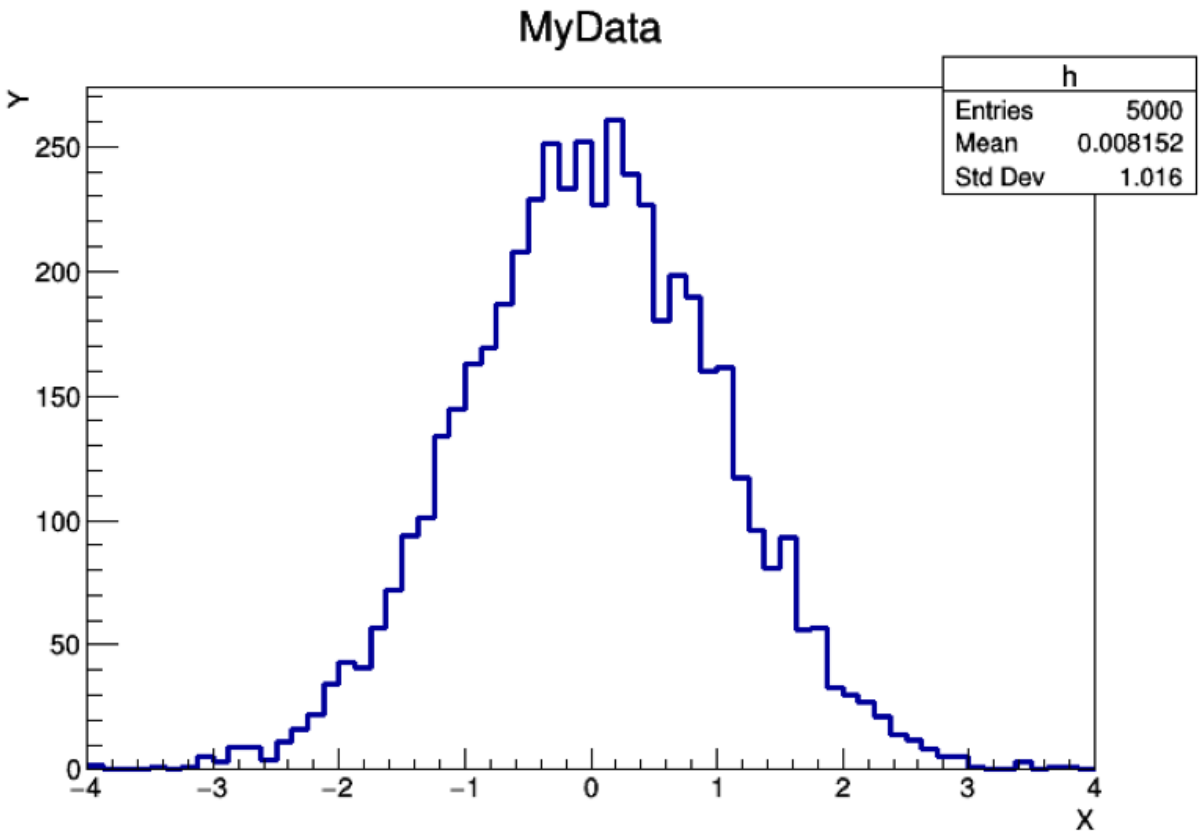
```
Welcome to ROOTaaS 6.05/01
```

In [2]:
```
%%cpp
auto myHisto = TH1F("h","MyData;X;Y",64,-4,4); // C++11
```

In [3]:
```
h = ROOT.myHisto # Find the variable back in Python!
h.FillRandom("gaus")
c = ROOT.TCanvas()
h.Draw()
c.Draw()
```

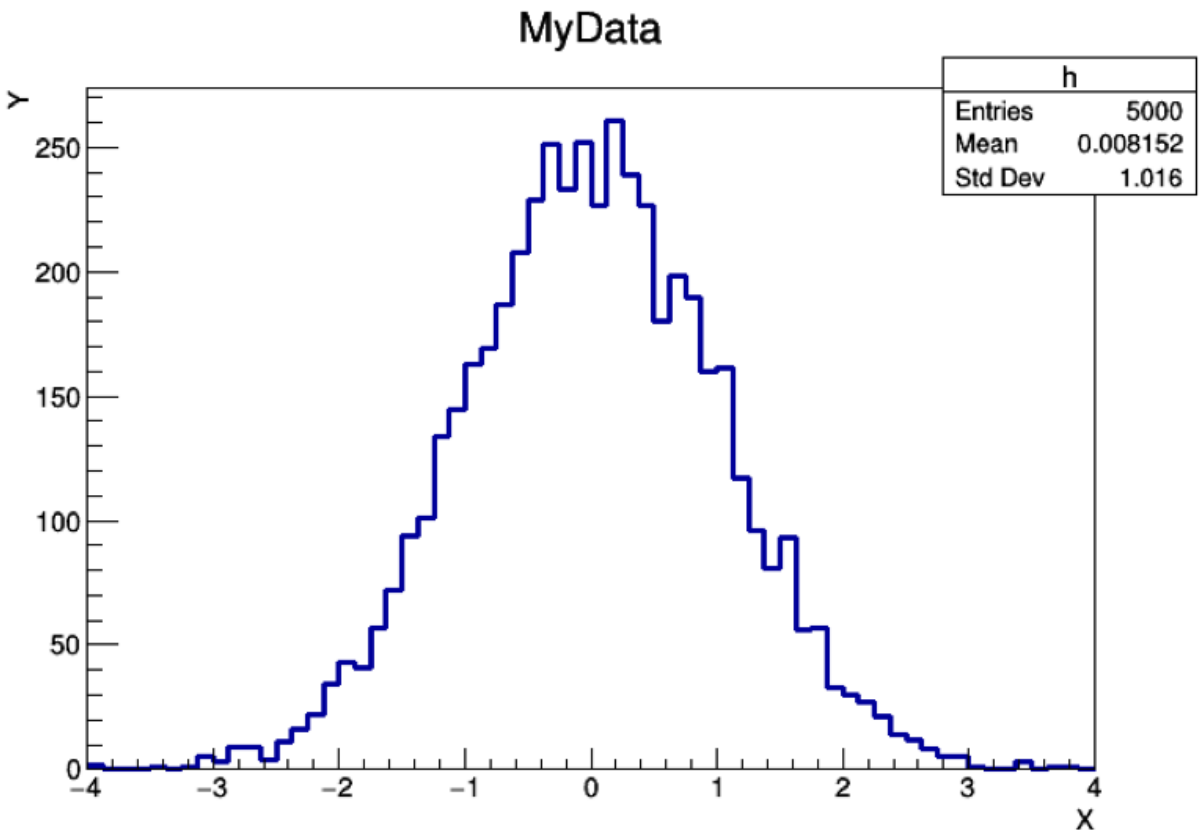
C++-Python Interoperability

```
c.Draw()
```



Seamless display of graphics

```
c.Draw()
```



Syntax Highlighting

In [4]:

```
%%cpp -d
double myG(double* x, double* par){
  auto res = (x[0]-par[1])/par[2];
  auto e = -.5 * res * res;
  return par[0] * exp(e); // declare function
}
```

In [4]:
```
%%cpp -d
double myG(double* x, double* par){
  auto res = (x[0]-par[1])/par[2];
  auto e = -.5 * res * res;
  return par[0] * exp(e); // declare function
}
```

In [5]:
```
f = ROOT.TF1("myGf",ROOT.myG,-5,5,3)
f.SetParameters(200,0,1);f.SetParNames("N","mu","sigma")
fr = ROOT.h.Fit(f,"S") # Capture printouts
```

In [4]:
```
%%cpp -d
double myG(double* x, double* par){
  auto res = (x[0]-par[1])/par[2];
  auto e = -.5 * res * res;
  return par[0] * exp(e); // declare function
}
```

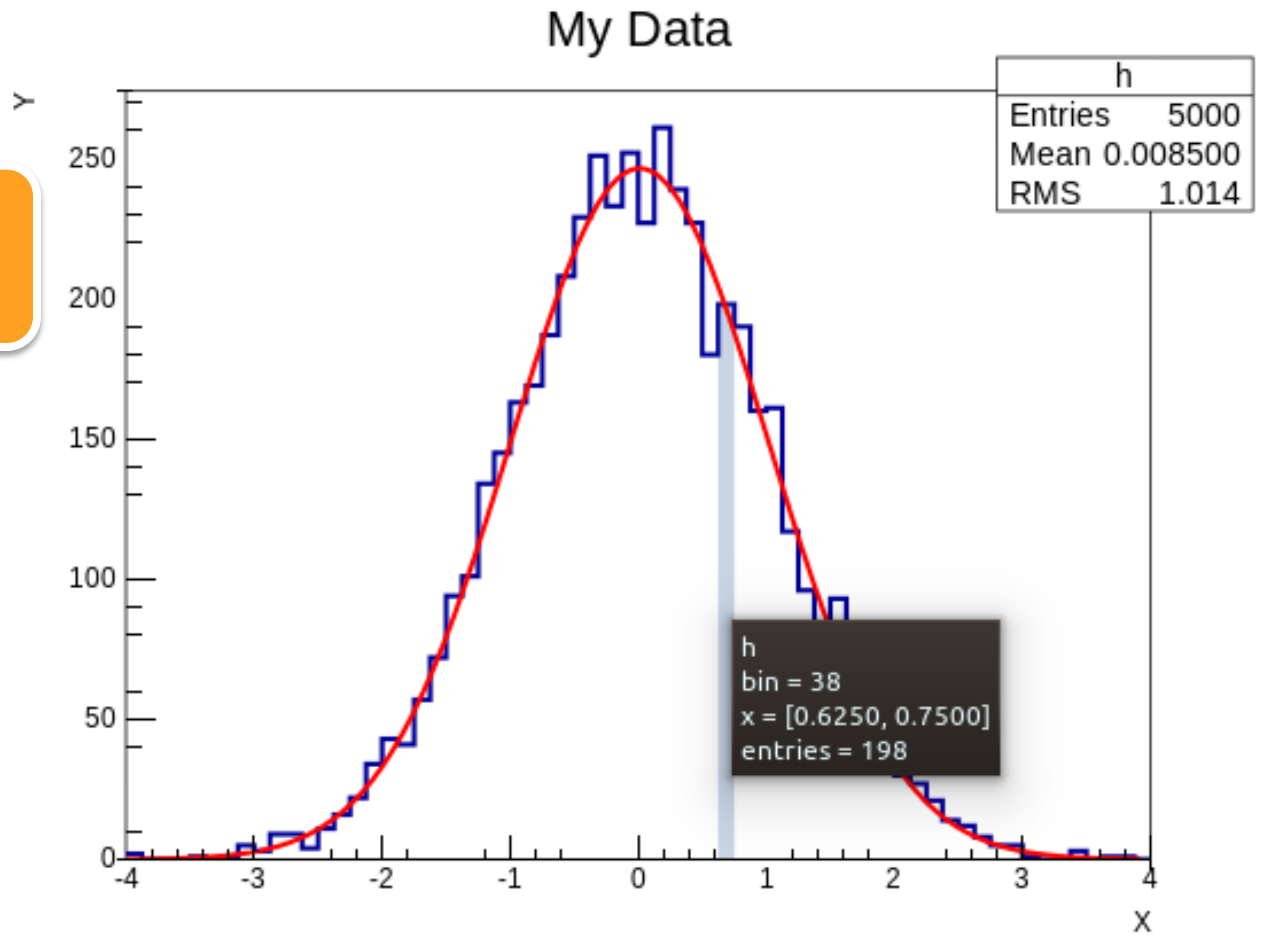
In [5]:
```
f = ROOT.TF1("myGf",ROOT.myG,-5,5,3)
f.SetParameters(200,0,1);f.SetParNames("N","mu","sigma")
fr = ROOT.h.Fit(f,"S") # Capture printouts
```

```
 FCN=47.4997 FROM MIGRAD    STATUS=CONVERGED      69 CALLS          70 TO
TAL
                     EDM=2.04372e-09    STRATEGY= 1      ERROR MATRIX ACC
URATE
  EXT PARAMETER                                   STEP         FIRST
  NO.   NAME      VALUE            ERROR          SIZE      DERIVATIVE
   1  N            2.46469e+02   4.31493e+00   1.19092e-02  -5.38026e-06
   2  mu           1.04793e-02   1.43576e-02   4.87640e-05   4.15093e-03
   3  sigma        1.00316e+00   1.03818e-02   2.86307e-05  -2.55310e-04
```

1	N	2.46469e+02	4.31493e+00	1.19092e-02	-5.38026e-06
2	mu	1.04793e-02	1.43576e-02	4.87640e-05	4.15093e-03
3	sigma	1.00316e+00	1.03818e-02	2.86307e-05	-2.55310e-04

In [6]:
```
ROOT.enableJSVis() # Not active by default yet!
c.Draw()
ROOT.disableJSVis()
```

JSROOT Visualisation



In [10]:
```
%%cpp -a
// Create dictionaries, a library and load it
#include <string>
class myClass{
 public:
  myClass(){};
  myClass(const char* name):fName(name){};
  const char* getName() const{return fName.c_str();}
 private:
  std::string fName = "";
};
```

```
Info in <TUnixSystem::ACLiC>: creating shared library
/home/rw15u099/PresentationNotebooks/e9c1711f_C.so
```

In [12]:
```
myObj = ROOT.myClass("theName")
ofile = ROOT.TFile("ofile.root","recreate")
h.Write()
ofile.WriteObjectAny(myObj,"myClass",myObj.getName())
ofile.Close()
```

In [13]:
```
%%bash
rootls -l ofile.root
```

```
TH1F      Sep 11 15:29  h         "MyData"
myClass   Sep 11 15:29  theName   "object title"
```

All the power of ROOT: Dictionaries, I/O, runtime loading of libraries

Take-away Message

"import ROOT" turns on all notebook goodies

- Tab-completion
- C++ cells, ACLiC
- Display of graphics
- Syntax highlighting

All the power of ROOT and the ROOT Python bindings, PyROOT, are there

Like before, but better

CERNBox Terminal Control Panel Logout

Files Running Clusters

Select items to perform actions on them.

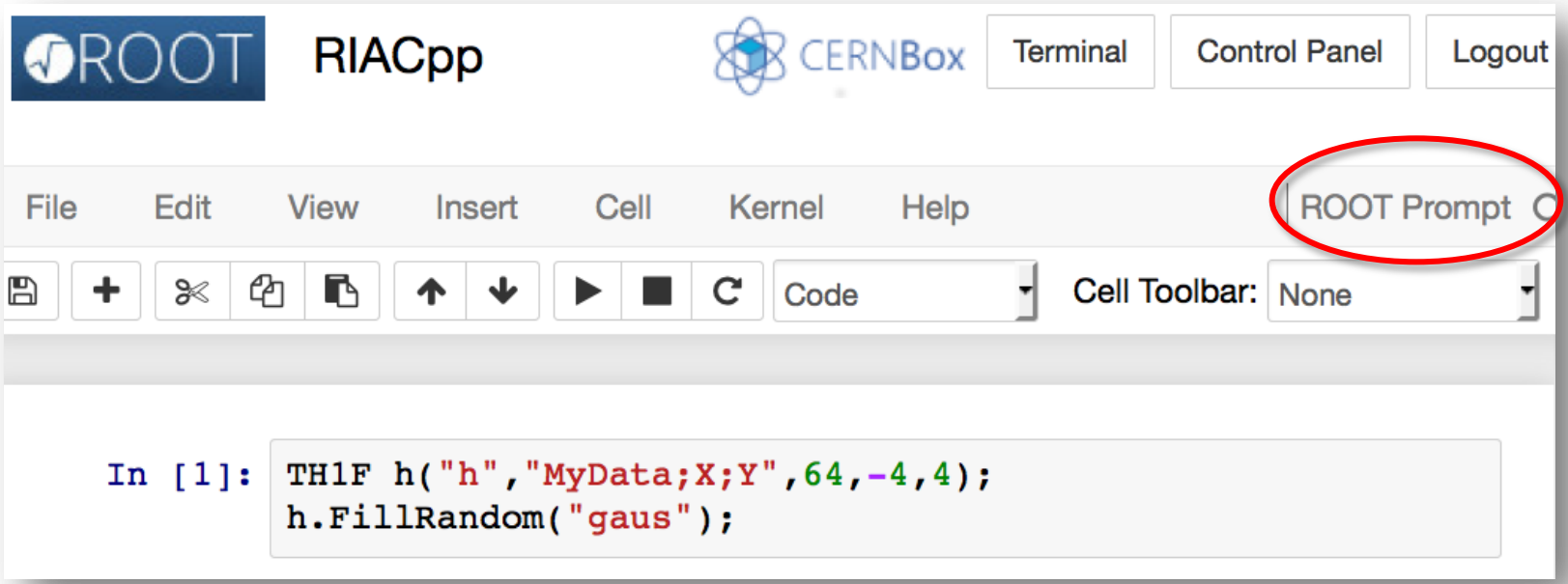
Upload New

Text File
Folder
Terminal

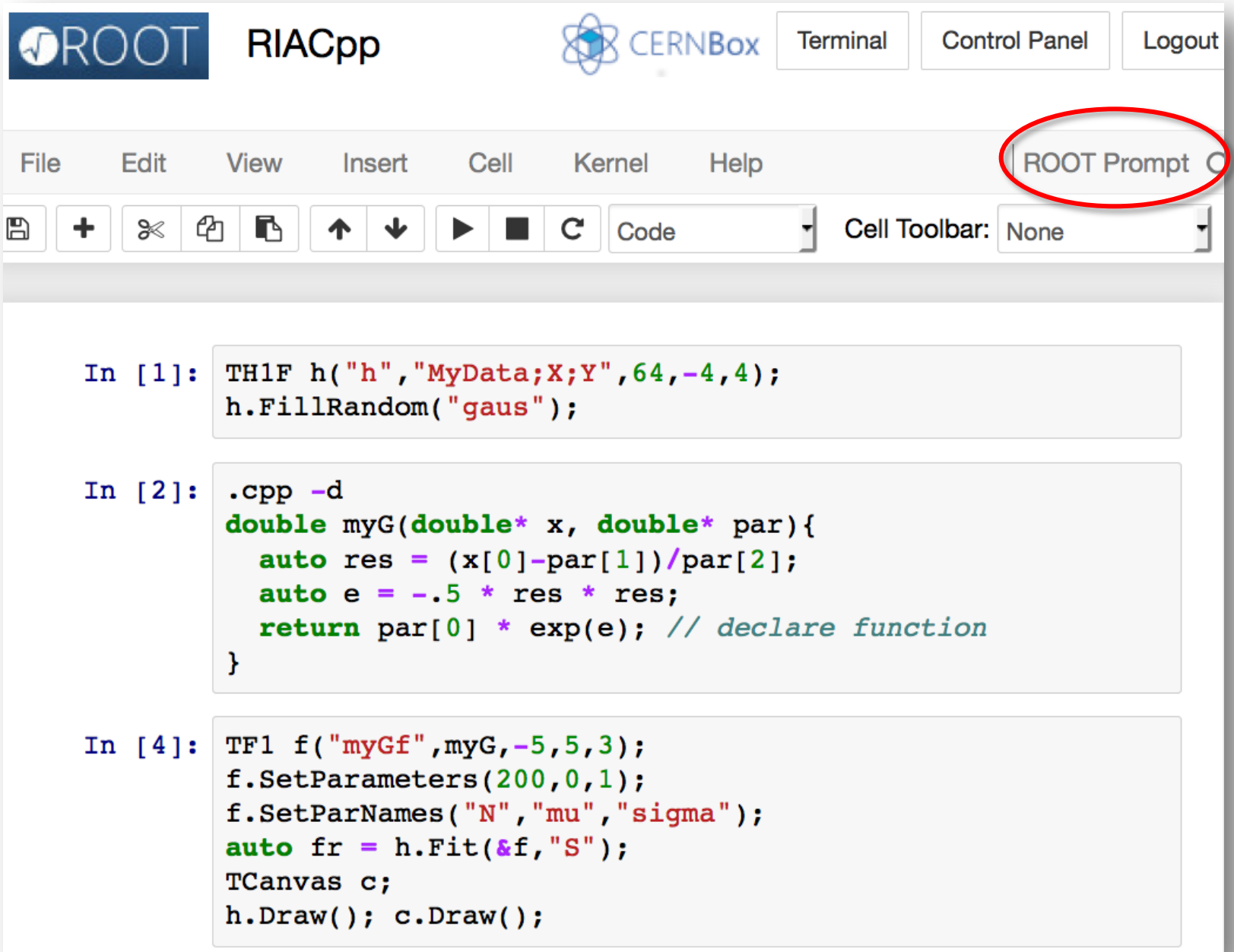
Notebooks
Python 2
Python 3
ROOT Prompt

PresentationNotebooks
cernbox
HowTo_ROOT-Notebooks.ipynb
HowTo_ROOT-Notebooks_Long.ipynb
My First Notebook.ipynb
Untitled.ipynb

A C++ Notebook


ROOT
RIACpp
CERNBox
Terminal
Control Panel
Logout
File
Edit
View
Insert
Cell
Kernel
Help
ROOT Prompt
Code
Cell Toolbar:
None
In [1]:
TH1F h("h","MyData;X;Y",64,-4,4);
h.FillRandom("gaus");

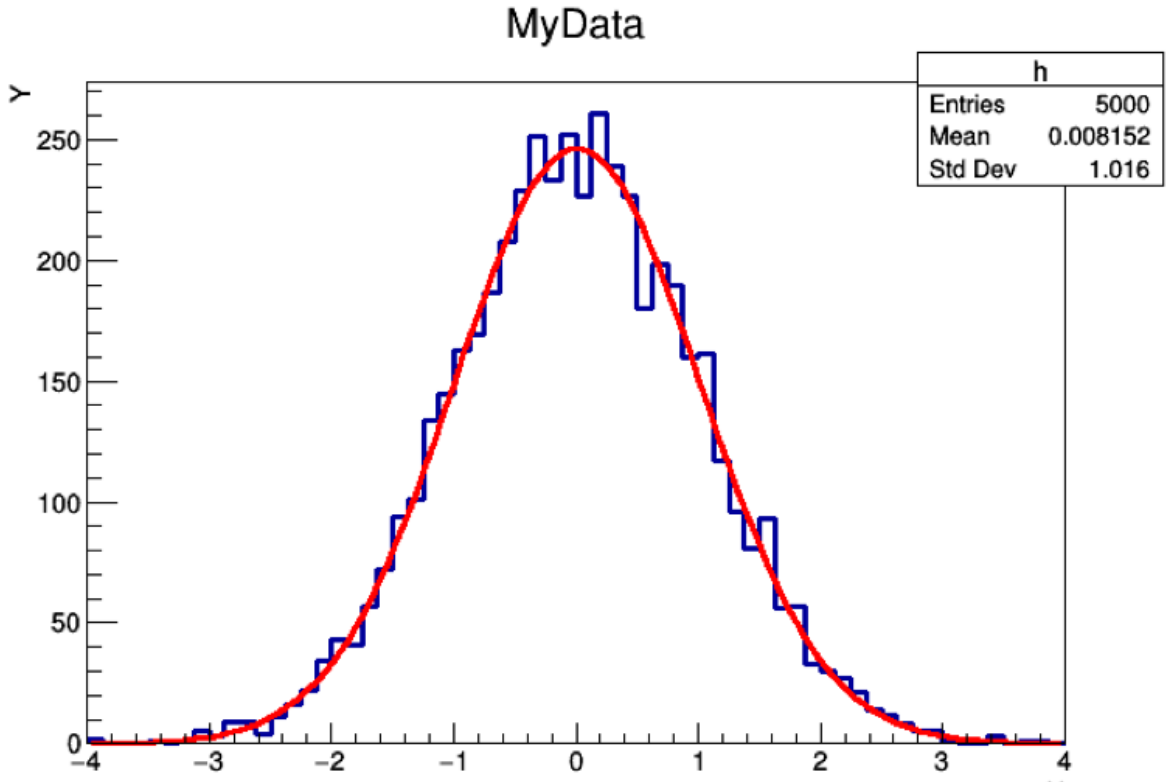
A C++ Notebook


ROOT
RIACpp
CERNBox
Terminal
Control Panel
Logout
File
Edit
View
Insert
Cell
Kernel
Help
ROOT Prompt
Code
Cell Toolbar:
None
In [1]:
TH1F h("h","MyData;X;Y",64,-4,4);
h.FillRandom("gaus");
In [2]:
.cpp -d
double myG(double* x, double* par){
auto res = (x[0]-par[1])/par[2];
auto e = -.5 * res * res;
return par[0] * exp(e); // declare function
}
In [4]:
TF1 f("myGf",myG,-5,5,3);
f.SetParameters(200,0,1);
f.SetParNames("N","mu","sigma");
auto fr = h.Fit(&f,"S");
TCanvas c;
h.Draw(); c.Draw();

A C++ Notebook

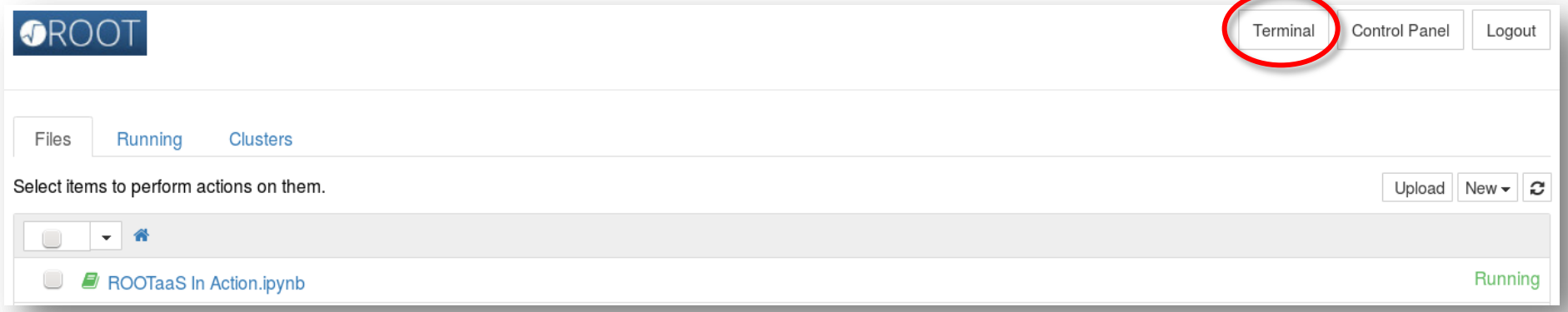
NO.	NAME	VALUE	ERROR	SIZE	DERIVATIVE
1	N	2.46469e+02	4.31493e+00	1.19092e-02	-5.38026e-06
2	mu	1.04793e-02	1.43576e-02	4.87640e-05	4.15093e-03
3	sigma	1.00316e+00	1.03818e-02	2.86307e-05	-2.55310e-04

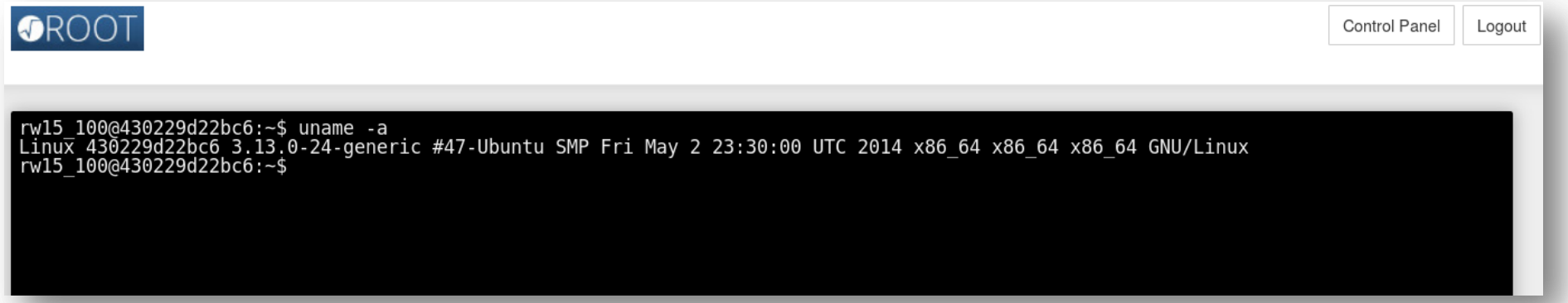
```
Info in <TCanvas::MakeDefCanvas>:  created default TCa
nvas with name c1
```



Going the Extra Mile

Make terminal available with one click!





Documentation, Documentation, Documentation

- **Examples** (15 already) from the *new* ROOT Tutorials can be found at:

https://root.cern.ch/code-examples#notebooks

both in Python and C++ (and mixed!)

- **"Howto"s**

"How To use ROOT in a Notebook" instructions and

"How To activate a *ROOT Prompt* kernel in Your IPython Notebook"

https://root.cern.ch/howtos#Language%20Bindings

- Notebook technology also adopted for writing the most "pragmatic" HowTos: https://root.cern.ch/howtos

Try It Out!

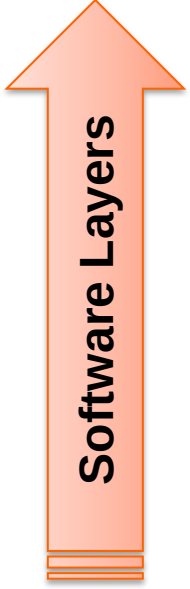
Install ROOT, install IPython notebooks and…

```
$ root --notebook
```

This command:

1. Starts a local notebook server
2. Connects to it via the browser

Provides a C++ notebook and the rest of ROOT notebook goodies



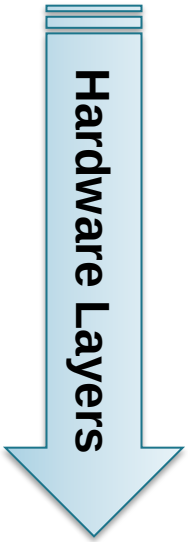
ROOT

iPyROOT
(ROOT-Notebooks integration)

JupyterHub +
CERN Add-ons

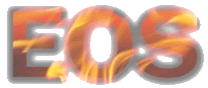
Storage

CPUs

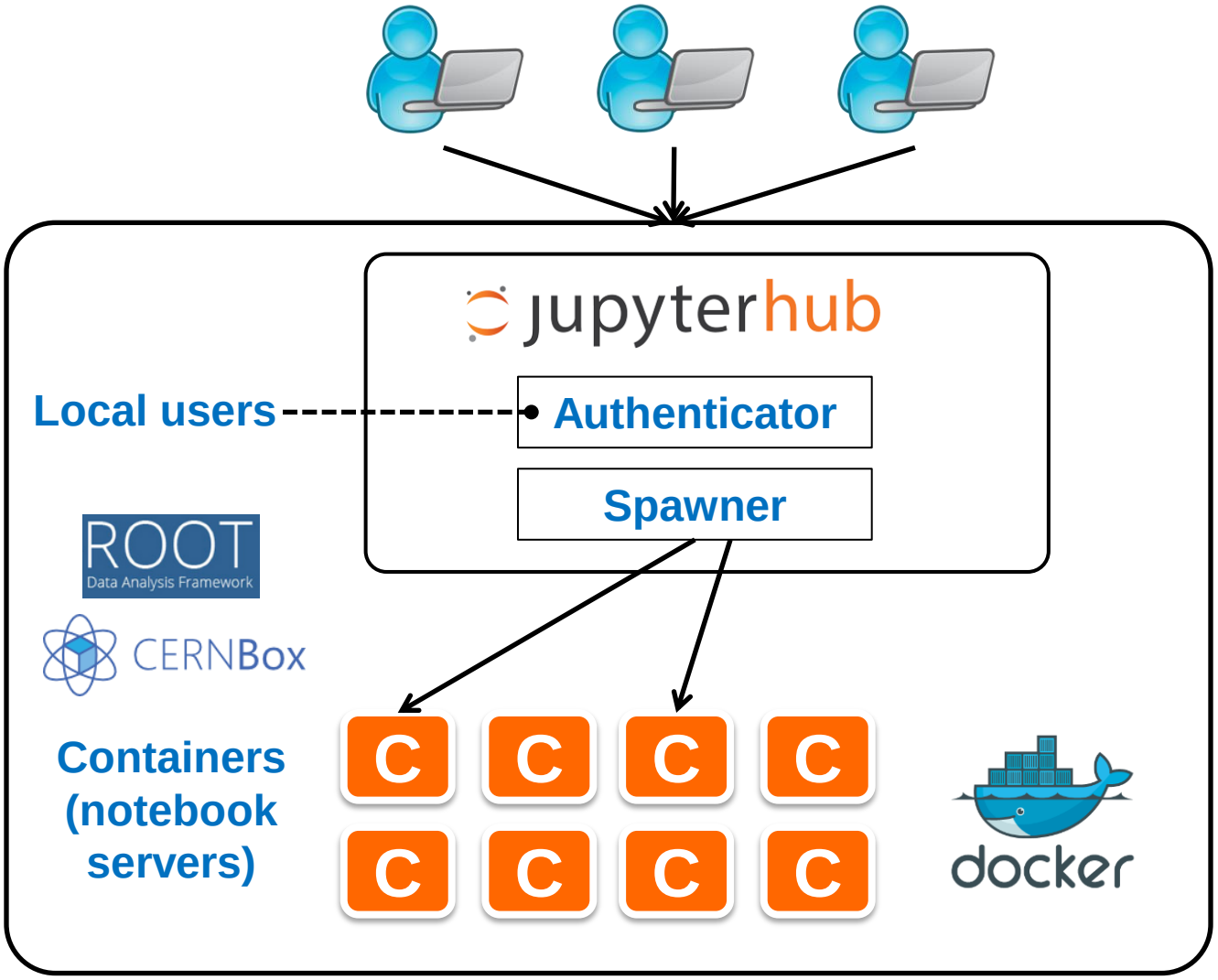


ROOTaaS and CERN services' Portfolio

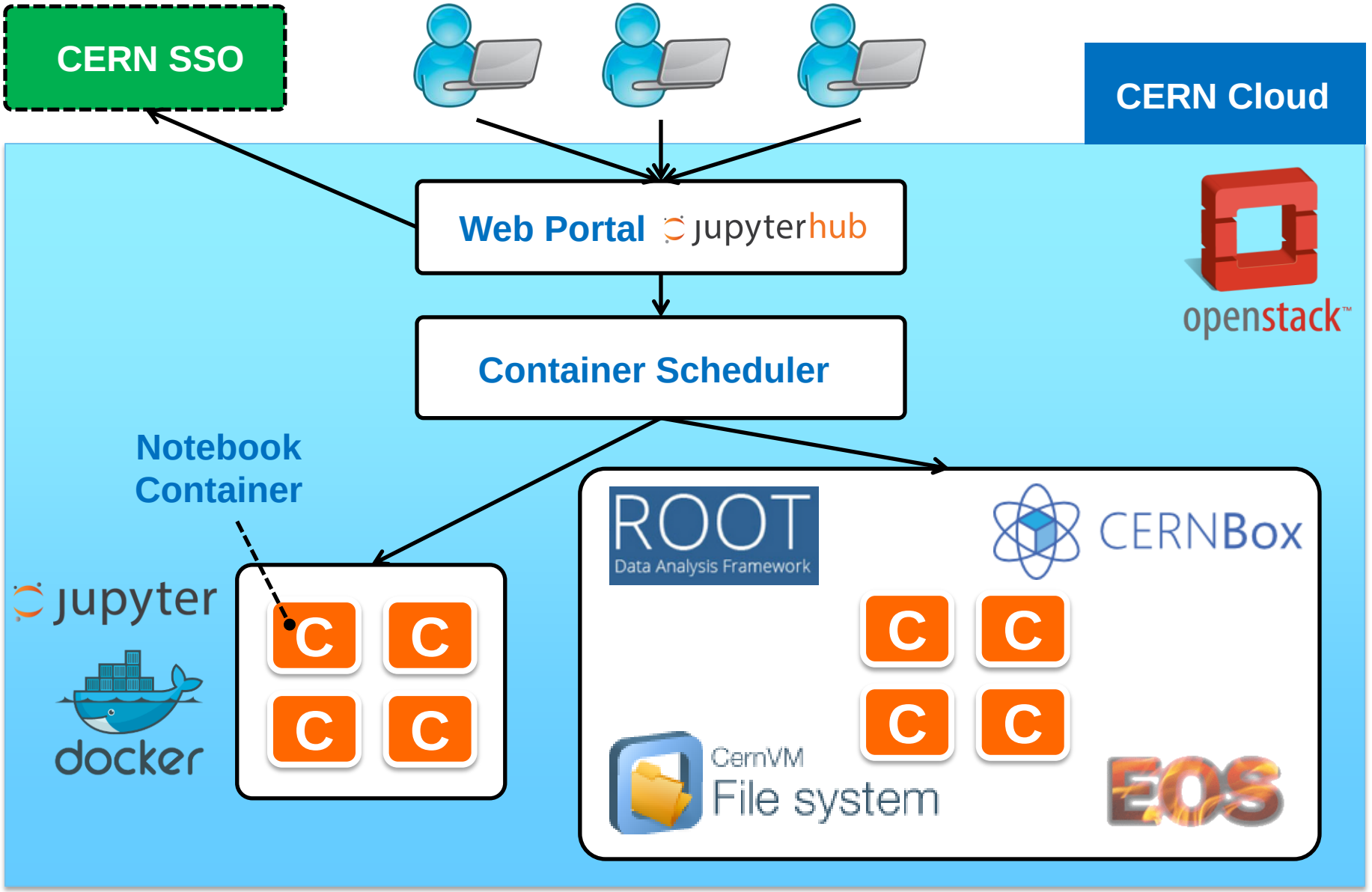
Integration With CERN's Ecosystem

- Centrally provide ROOT as a Service
- Authentication with CERN credentials
- Connect to virtual machines in the OpenStack Cloud
- Software distribution: CVMFS 
- Storage access: CERNBox, EOS  
  - All data potentially available!
- Synergy with document Sharing (e.g. CERN Indico)
  - Notebook visualiser available in the next Indico release, with ROOTaaS C++ highlighter integrated 

What we have now: Demo Server



Pilot Service Proposal



Potential “Daily Use-Case”

- Launch jobs on the batch farm
- Access notebook on a VM in the OpenStack instance
- Inspect produced data via CERNBox/EOS from the notebook
- Create plots and output data
- Share, access plots (and output data!) on the web with CERNBox web interface / Indico
- Security and confidentiality guaranteed by the usual CERN standards

Conclusions

- ROOT is now integrated with notebooks
  - Python and C++ interactive shells
  - Tab completion, C++/Python integration, syntax highlighting, graphics inlining, shell commands
  - Available now (6.05/02)!
- Integration with the CERN services portfolio
  - Collaborating with IT department
    - Special thanks to M. Lamanna, J. Moscicki and L. Mascetti
  - Work in progress, usable demo server available
  - Draft pilot service proposal under evaluation

Backup slides

Integration: Main Requirements

- Code in macros/programs usable in notebooks (and vice versa)
- Provide a novel ROOT Prompt (C++) kernel
  - A notebook which is a web based ROOT prompt
- Easy access to well known ROOT and notebooks features
- Provide clear, useful examples and documentation

Requirements satisfied
Delivered in release 6.05/02

Now it's time to take a tour of the new provided functionalities!

Chosen Technology

Jupyterhub: manages login of users and redirection to notebook

- Existing solution: https://github.com/jupyter/jupyterhub
- Allows encapsulation: spawn Linux container at logon
  - User isolated from the host, modulo volumes explicitly mounted (cvmfs, CERNBox)
- Needs to be customised, e.g.:
  - CERN sign-on procedure
  - Docker image for the container

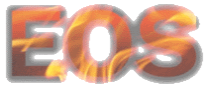
A Success Story

- CERN Summer Student Program
- >100 Students hosted at CERN for 8-13 weeks
- Internship + Lectures program
  - ROOT Tutorial for students organised, 4 sessions

Last session of the tutorial: interactive notebooks offered

- Single 24 cores box, Beta version of the software layer
- **50 participants, perfect scaling, a success!**
  - https://indico.cern.ch/event/407519

Glossary

EOS 

Disk-based low latency storage infrastructure for physics users. Main target: physics data analysis.

CERNBox 

Functionality analogous to Dropbox™. Synchronisation capabilities between user machines and central repository. Data stored on EOS.

Indico 

Manage complex conferences, workshops and meetings.

CVMFS 

HTTP based network FS, optimized to deliver experiment software

Files aggressively cached and downloaded on demand.

The Demo

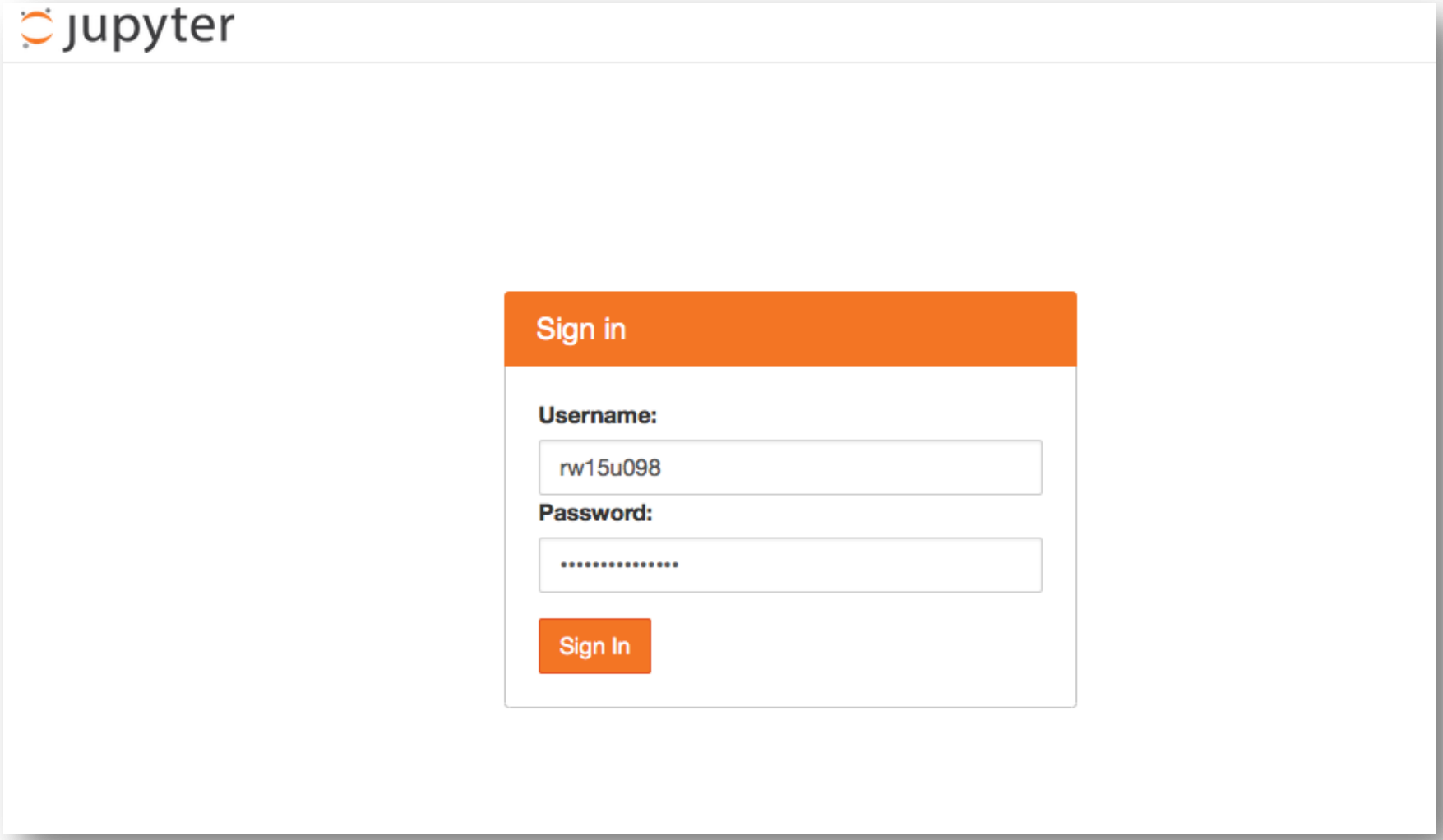
Live Demo

Time to go back see this workflow in action!

We will:

- Create a simple plot and a ROOT file with ROOTaaS
- Share it with CERNBox

Live Demo



Live Demo

Terminal | Control Panel | Logout

Files | Running | Clusters

Select items to perform actions on them.

Upload | New

cernbox

Live Demo

Terminal | Control Panel | Logout

Select items to perform actions on them.

Upload | New | 

/ cernbox

- ..
- tutorials
- HowTo_ROOT-Notebooks.ipynb
- My First Notebook.ipynb

Live Demo

My First ROOT Notebook

This is an example that aims to show the capabilities of ROOT once integrated in a notebook.

In [1]:
```
import ROOT
```

```
Welcome to ROOTaaS 6.05/01
```

In [2]:
```
h = ROOT.TH1F("myHisto","My Title!;My X Axis;My Y Axis",64,-4,4)
h.FillRandom("gaus")
```

In [3]:
```
c = ROOT.TCanvas("myCanvas","myCanvasTitle",1024,768)
h.Draw()
c.Draw()
```

Live Demo

This is an achievement. Let's save this plot and the histogram itself in a ROOT file.

In [4]:
```
c.Print("myPlot.pdf")
```

```
Info in <TCanvas::Print>: pdf file myPlot.pdf has been created
```

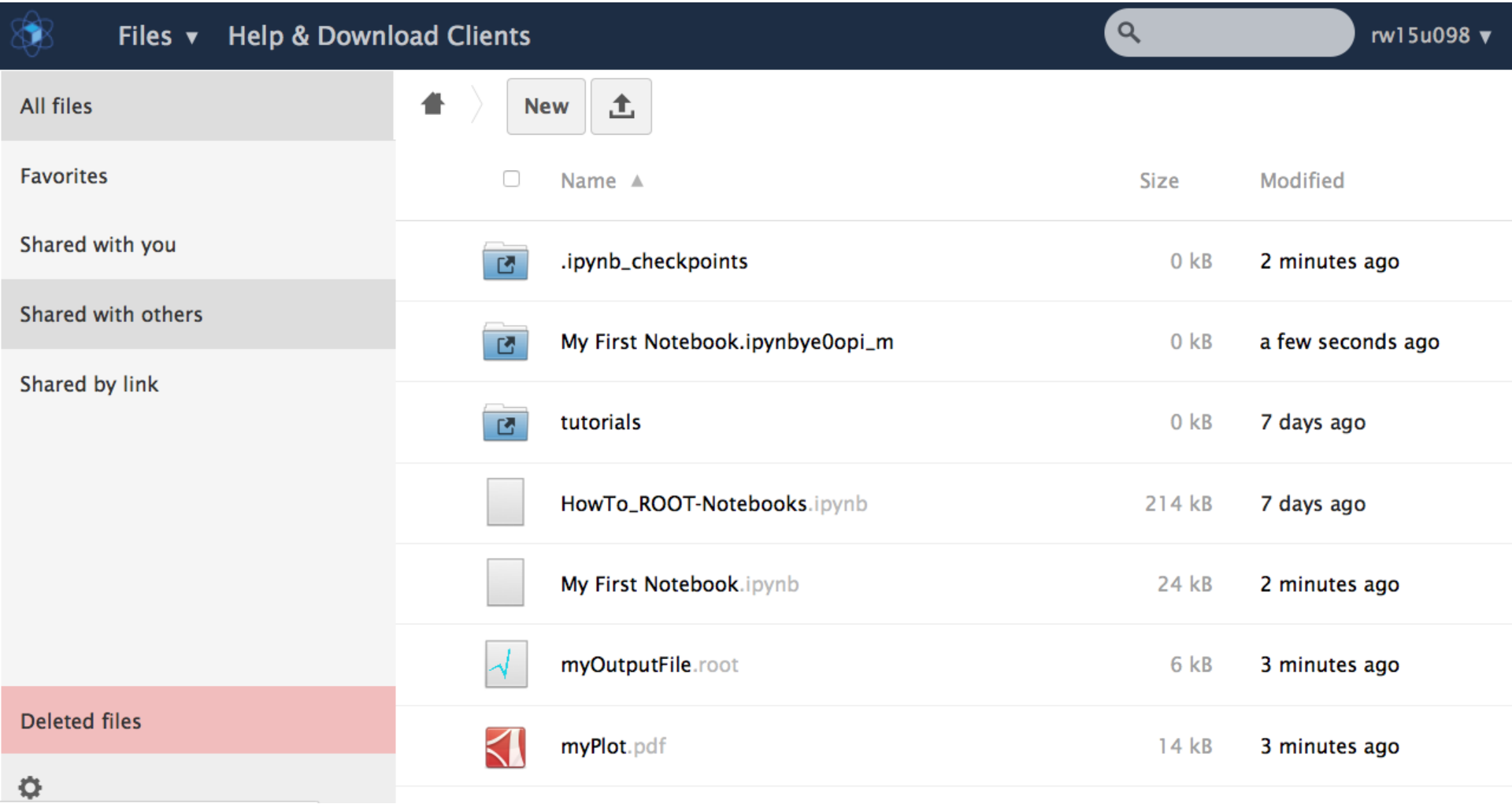
In [5]:
```
ofile = ROOT.TFile.Open("myOutputFile.root","recreate")
h.Write()
ofile.Close()
```

In [6]:
```
%%bash
ls
```

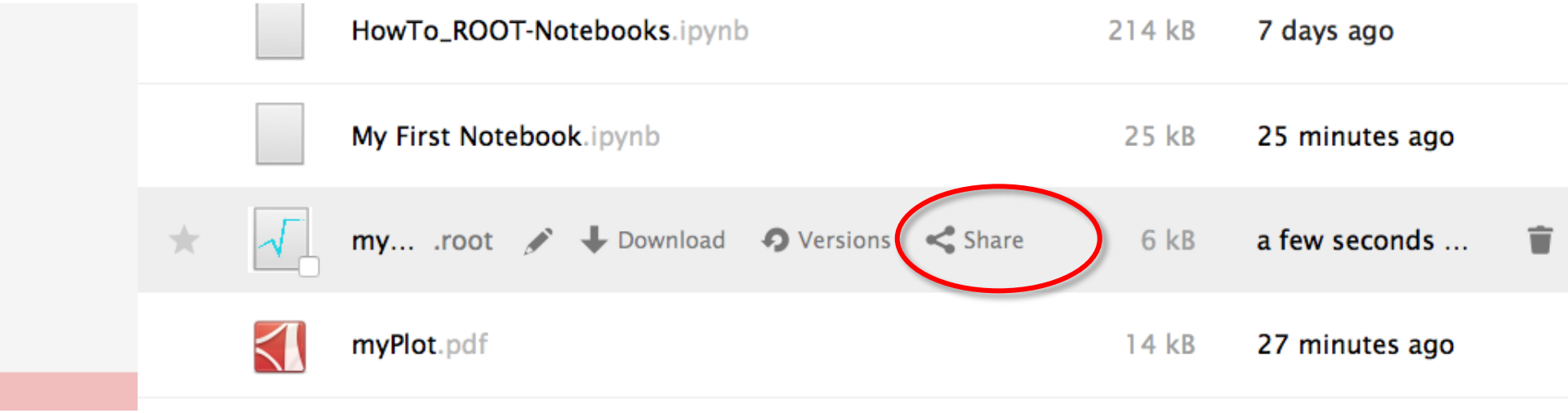
```
HowTo_ROOT-Notebooks.ipynb
My First Notebook.ipynb
myOutputFile.root
myPlot.pdf
tutorials
```

Now go and check on the **CERNBOX** web interface **your data**!

Live Demo


Files
Help & Download Clients
rw15u098
All files
Favorites
Shared with you
Shared with others
Shared by link
Deleted files
New
Name
Size
Modified
.ipynb_checkpoints 0 kB 2 minutes ago
My First Notebook.ipynbye0opi_m 0 kB a few seconds ago
tutorials 0 kB 7 days ago
HowTo_ROOT-Notebooks.ipynb 214 kB 7 days ago
My First Notebook.ipynb 24 kB 2 minutes ago
myOutputFile.root 6 kB 3 minutes ago
myPlot.pdf 14 kB 3 minutes ago

Live Demo


HowTo_ROOT-Notebooks.ipynb 214 kB 7 days ago
My First Notebook.ipynb 25 kB 25 minutes ago
my... .root Download Versions Share 6 kB a few seconds ...
myPlot.pdf 14 kB 27 minutes ago

Try ROOTaaS Now!

- Get ROOT, try it in a notebook on your laptop or…
- Access the demo server:

www.cern.ch/rootaasdemo

Get a ROOTaaS account now (talk to Enric or Danilo)!

- Take a look to the provided notebooks, modify them, run them
  - Produce results!
  - Access data and plots via CERNBox (https://cernbox.cern.ch)
  - Develop locally, sync directory, run your code in the notebook
  - Share with others results and more

Thanks to the IT-DSS group, in particular *L. Mascetti, K. Moscicki and M. Lamanna* for their fundamental contribution to the creation of this demo!