

Introduction to SWAN

Diogo Castro, Enric Tejedor, Pedro Maximino
On behalf of the SWAN team

<https://cern.ch/swan>



November 6th, 2024

CERN School of Computing on IT Services





SWAN in a nutshell

- › Interactive analysis with a web browser
 - No local installation is needed
 - Based on Jupyter Notebooks
 - Calculations, input data and results “in the Cloud”
- › Good for data analysis and exploration, but also for teaching
- › Easy sharing of scientific results: plots, data, code
- › **Added value: integration with CERN infrastructure and services!**



Integrating (CERN) services





The Notebook

Text

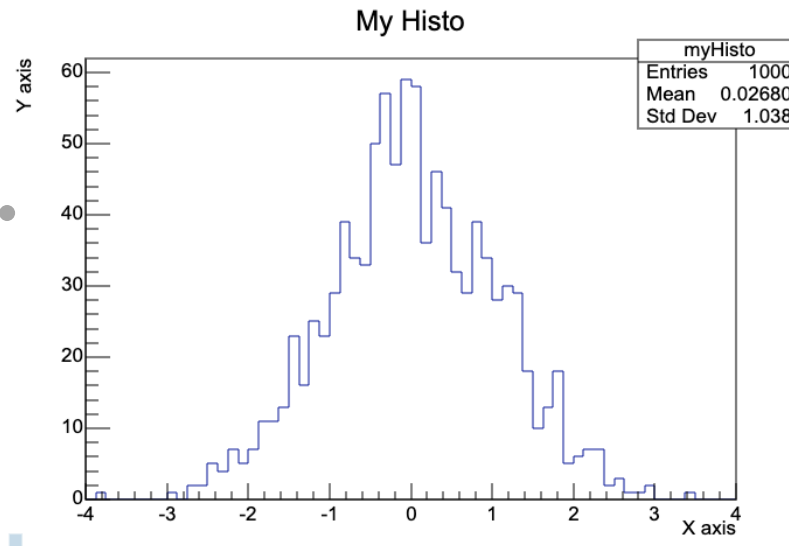
Code

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()
```



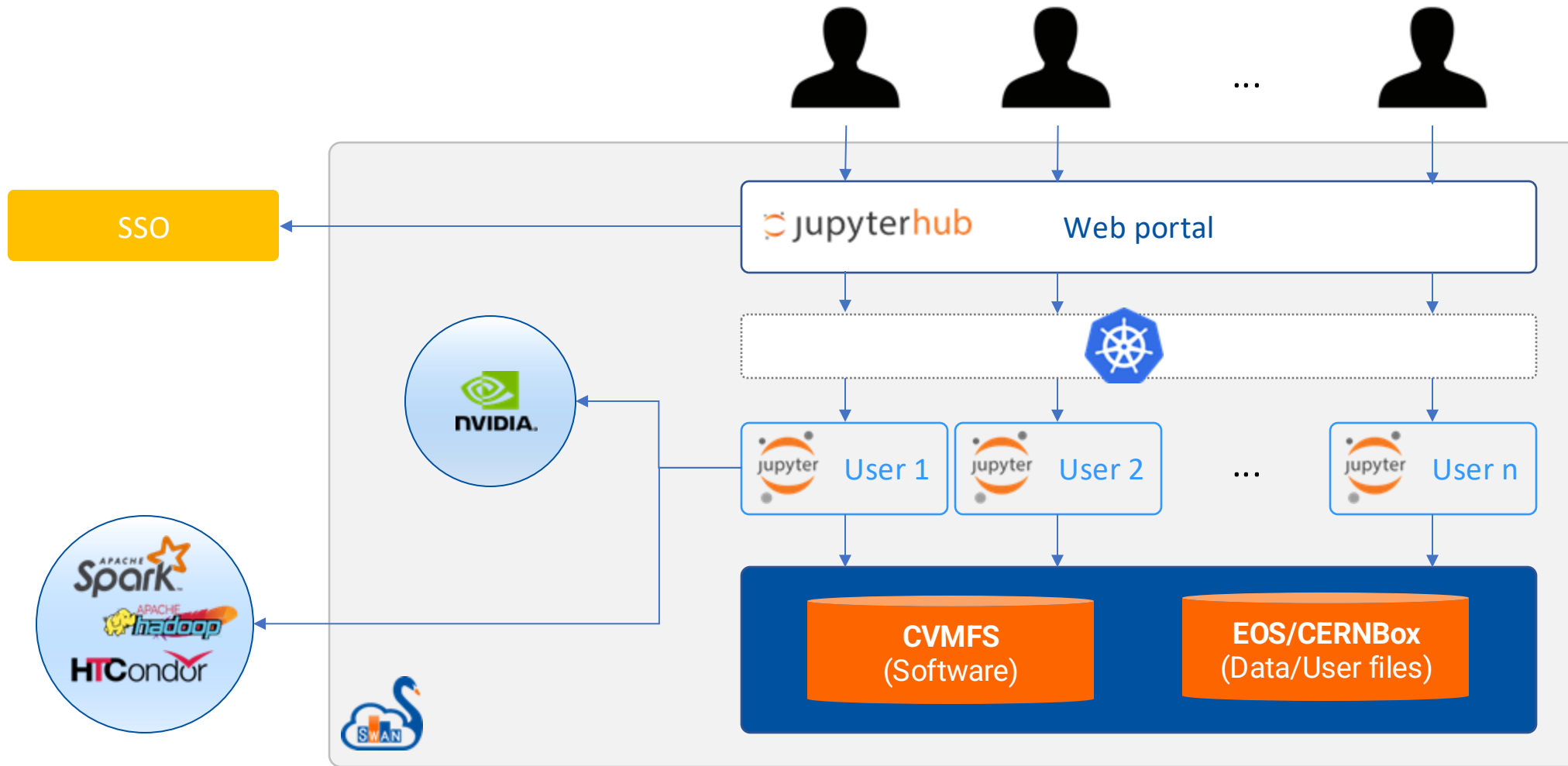
We'll try now to beautify the plot a bit, for example filling the histogram with a colour and setting a grid on the canvas.

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





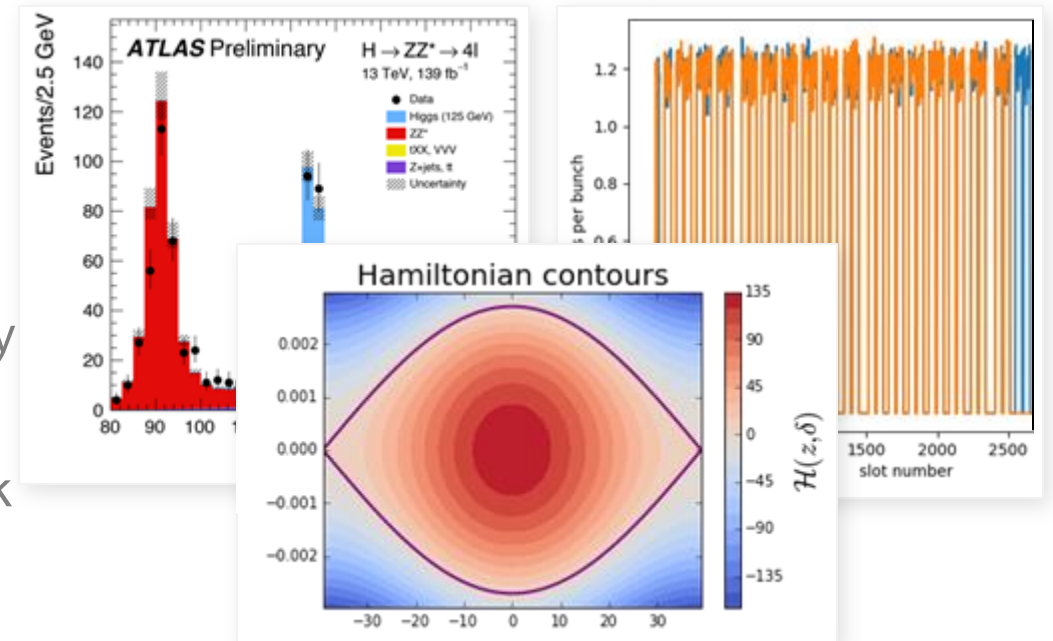
Architecture





Main user communities

- > Physics analysis
 - Usually last stages of analysis
 - Interactive, exploratory
 - Collision event data, ntuple-like, columnar
 - More and more with Machine Learning
- > Non-physics analysis (e.g. ATS)
 - LHC studies: extract machine measurements, query machine settings
 - Beam dynamics simulation
 - Query and process LHC logs distributedly via Spark
 - Query and plot monitoring data in experiment DAQ systems
- > Education
 - Many schools/workshops use SWAN for teaching

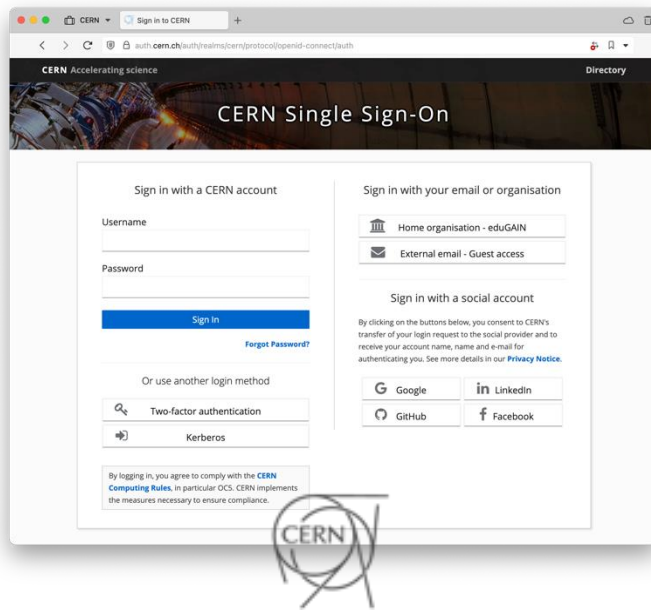


How to use it?

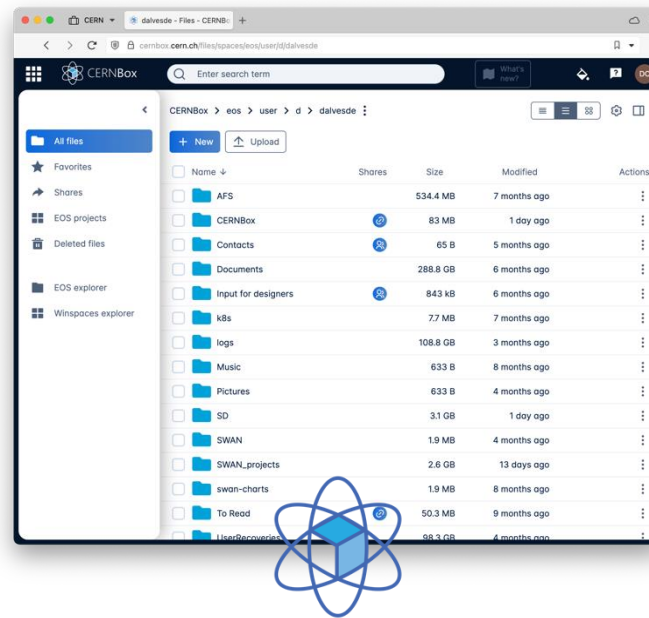


How to connect?

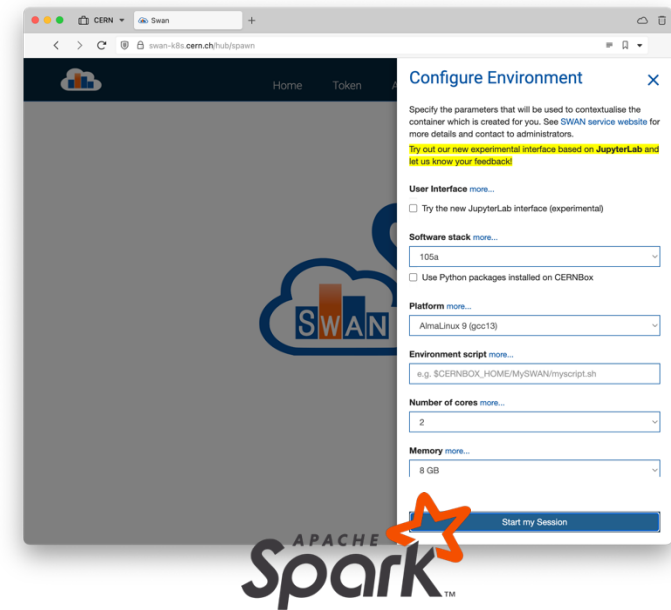
swan.cern.ch



Have a CERN account



Have a CERNBox space
(Open the service at least once!)



Optional:
Request access to Spark
([Service Now Request](#))





Start your session

Software Stack: what packages(versions) do you want?

<http://lcginfo.cern.ch/>

Personalise the environment? startup script!

Spark & HTCondor: do you want to offload computations to a CERN cluster? Which one?

Configure Environment

Specify the parameters that will be used to contextualise the container which is created for you. See [SWAN service website](#) for more details and contact to administrators.

Try out our new experimental interface based on JupyterLab and let us know your feedback!

User interface more...

Try the new JupyterLab interface (experimental)

Software stack more...

105a

Use Python packages installed on CERNBox

Platform more...

AlmaLinux 9 (gcc13)

Environment script more...

e.g. \$CERNBOX_HOME/MySWAN/myscript.sh

Number of cores more...

2

Memory more...

8 GB

External computing resources

Spark cluster more...

None

HTCondor pool more...

None

JupyterLab: Use the new JupyterLab interface?

CERNBox: Include installed packages in Python path

Platform: what system/compiler?

How much **memory**? And **cores**?





Classic UI

The screenshot shows the Classic UI interface for a Jupyter Notebook. The notebook title is "1 A Study about Cinemas in Canton Geneva". Below the title, it says "Based on a Demo by Danilo Piparo". The notebook content includes a section "1.1 Prepare the dataset" with a dataset URL and a code cell containing a shell command: `cat ga-s-1482810000_101.csv | grep Geneva | awk '{OFS=" "}' | sort -n`. A "Spark clusters connection" dialog box is open, showing options for connecting to "analytix". The dialog includes a section for "Bundled configurations" with checkboxes for various options like "Include SparkMetrics options", "Include S3FileSystem options", etc. A "Connect" button is at the bottom of the dialog.

The screenshot shows the "Share" page for a project named "SWAN". The page title is "SWAN > Share". Below the title, it says "Projects shared with me". There is a table listing shared projects with columns for NAME, SIZE, SHARED BY, and DATE.

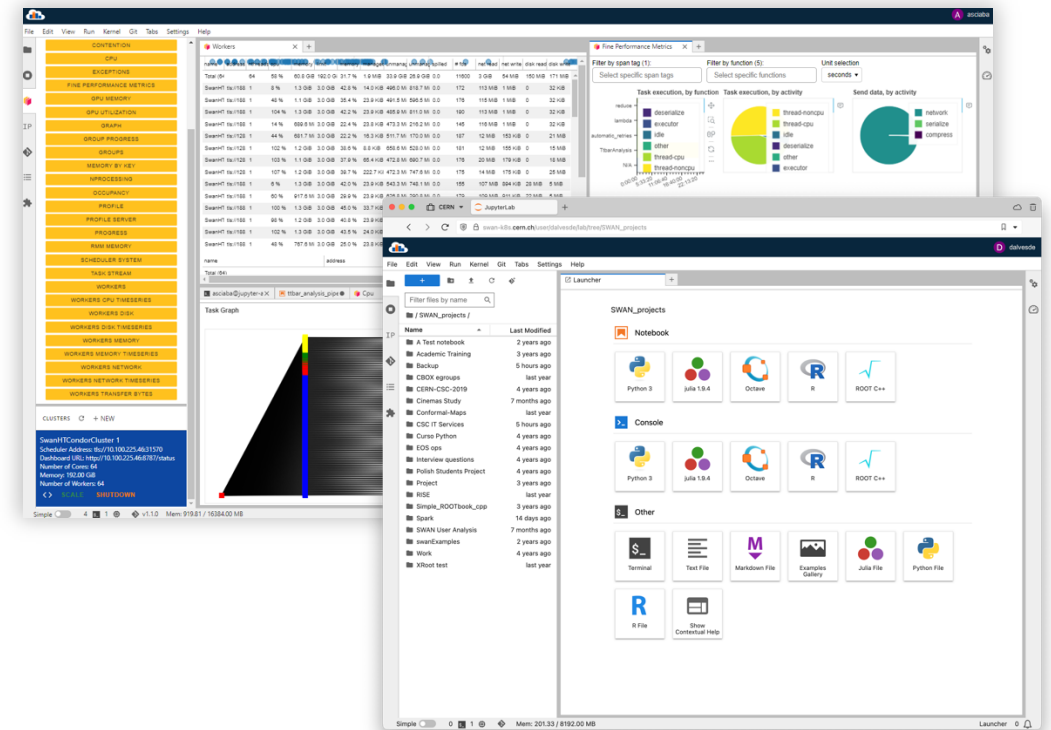
NAME	SIZE	SHARED BY	DATE
Academic Training2	673 kB	diccas	3 years ago
AFS_discolists	2.71 MB	lan	8 year ago
CERNBox	209 kB	lpresti	8 year ago
CondorGSoCTest	33.5 kB	moscicki	6 years ago
ITTF user statistics	342 kB	etejedor	9 days ago
sloexlab	2.24 GB	retaylor	2 years ago
Spark-DatROOT	236 kB	etejedor	6 years ago
Super Real Analysis with TOTEM data	2.29 kB	juystemon	5 years ago





JupyterLab

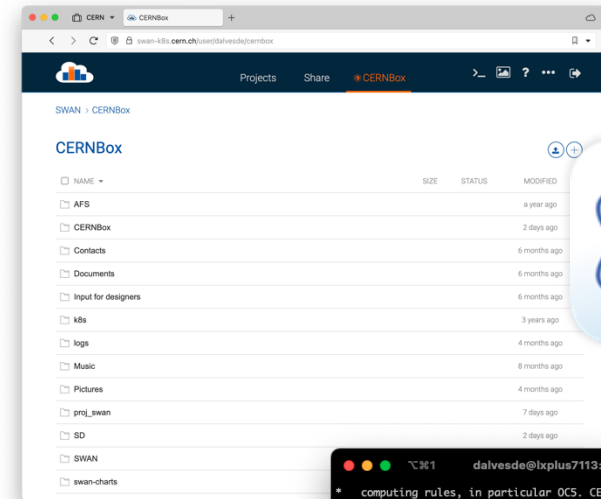
- The latest interface proposed by Project Jupyter
 - Notebooks, terminals, ...
 - ... and virtually anything via extensions
- Started if users tick the box in the form
 - Will initially coexist with classic UI
 - To be made the default in the future
- Most SWAN extensions already migrated to JupyterLab





CERNBox is your home

- > All the data our users need for their analysis
 - CERNBox as home directory
 - Experiment repositories, projects, open data, ...
- > Sync & Share
 - Files synced across devices and the Cloud
 - Simple collaborative analysis
- > Data accessible from other services
 - E.g. Ixplus



share



sync



```
dalvesde@lxplus7113:/eos/user/dj/dalvesde
* computing rules, in particular OCS. CERN implements
* the measures necessary to ensure compliance.
* https://cern.ch/ComputingRules
* Puppet environment: production, Roger state: production
* Foreman hostgroup: lxplus/nodes/Login
* Availability zone: cern-geneva-c
* LXPLUS Public Login Service - http://lxplusdoc.web.cern.ch/
* A C8 based lxplus8.cern.ch is now available
* Please read LXPLUS Privacy Notice in http://cern.ch/go/TrpV7
* .....
[dalvesde@lxplus7113 ~]$ cd /eos/user/dj/dalvesde
[dalvesde@lxplus7113 dalvesde]$
```





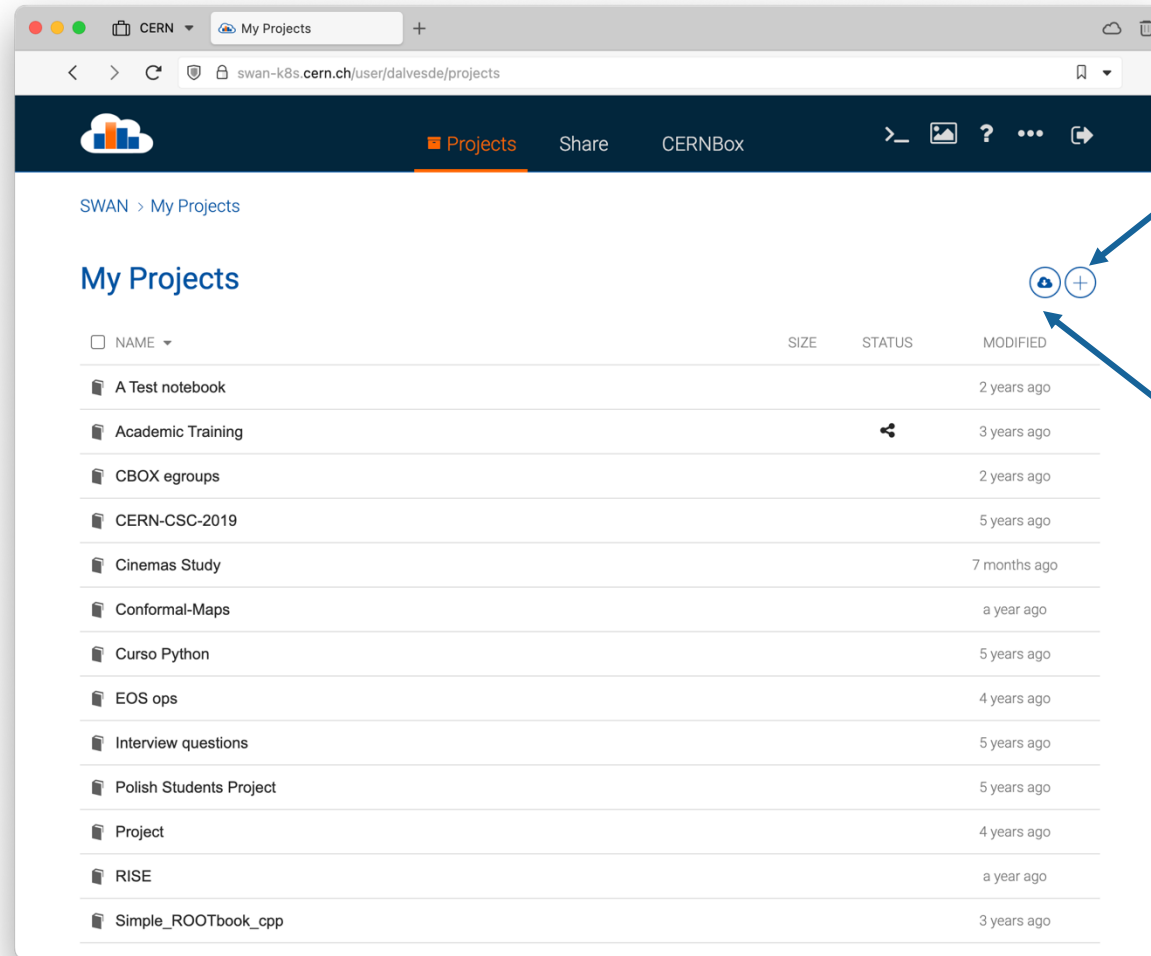
Projects

Folder with a set of notebooks, identified by a “.swanproject” file inside it

3.



<https://cern.ch/swanserver/cgi-bin/go?projurl=<path to your repo>>



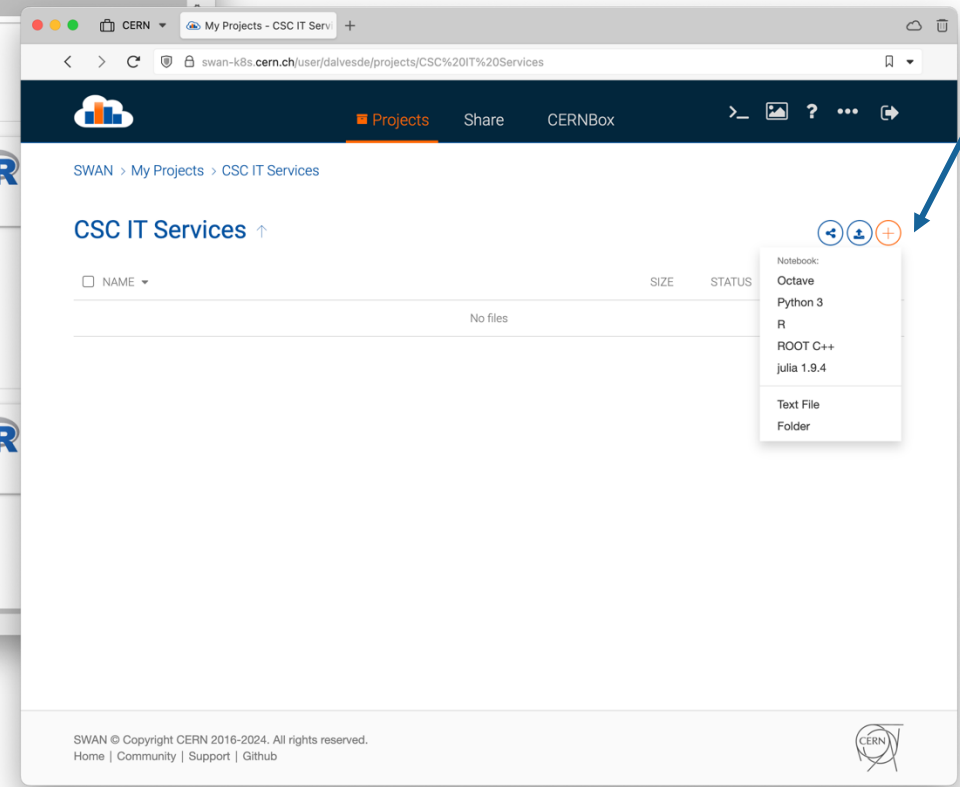
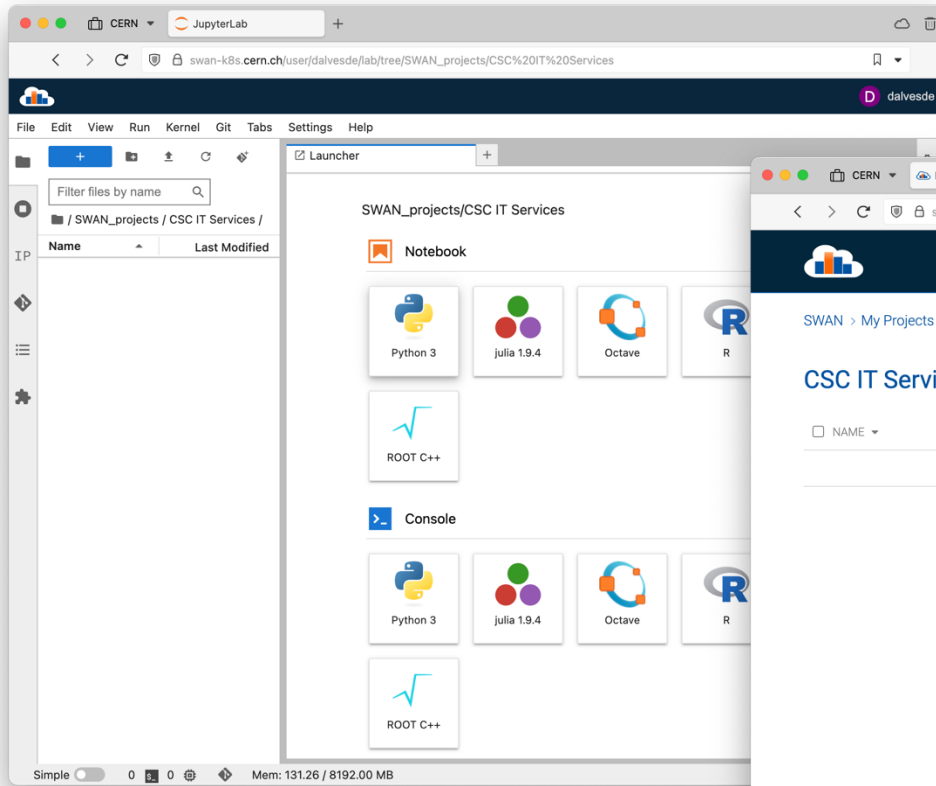
1. Create empty

2. Download from our Gallery, GitHub, CERN GitLab, CERNBox or ROOT website





Creating notebooks

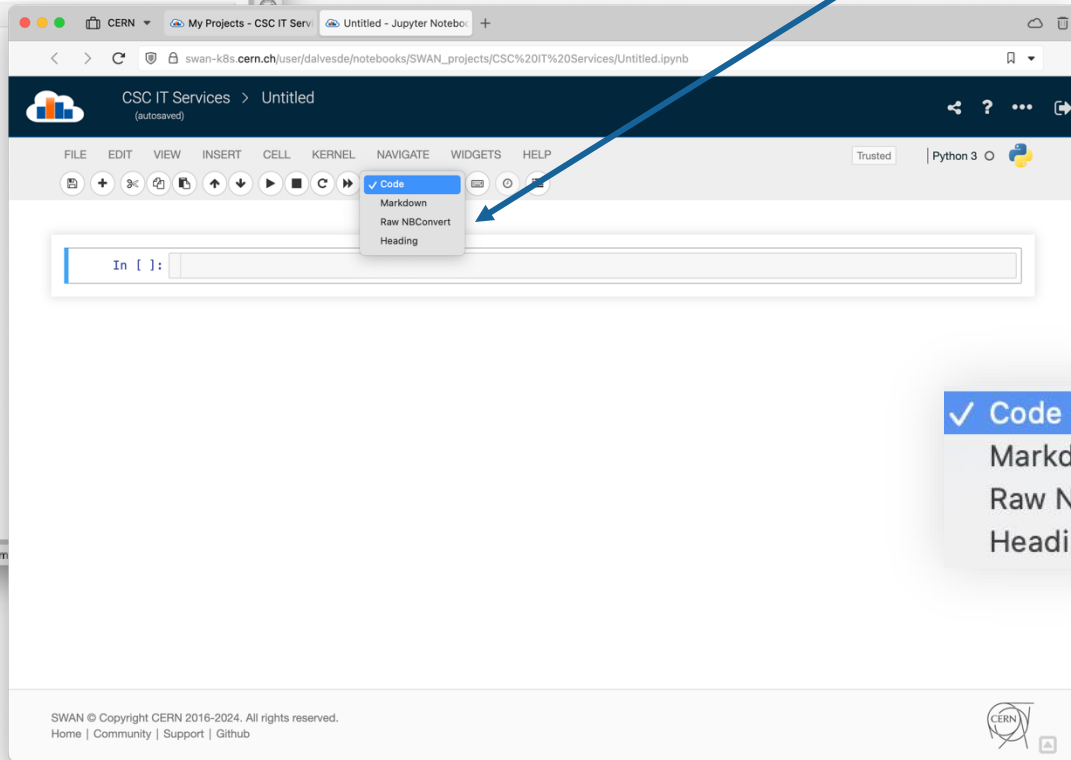
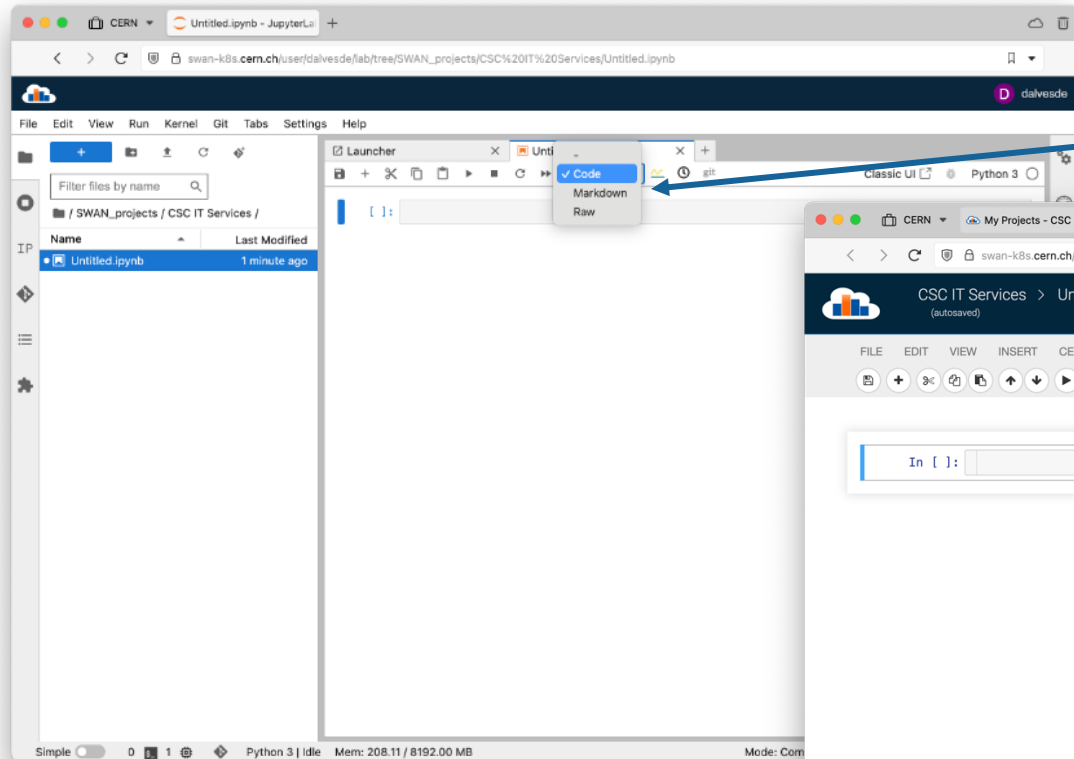


Click + and choose language





Notebook cells



Click to change the current cell type

- ✓ Code
- Markdown
- Raw NBConvert
- Heading





Saving notebooks

Autosave

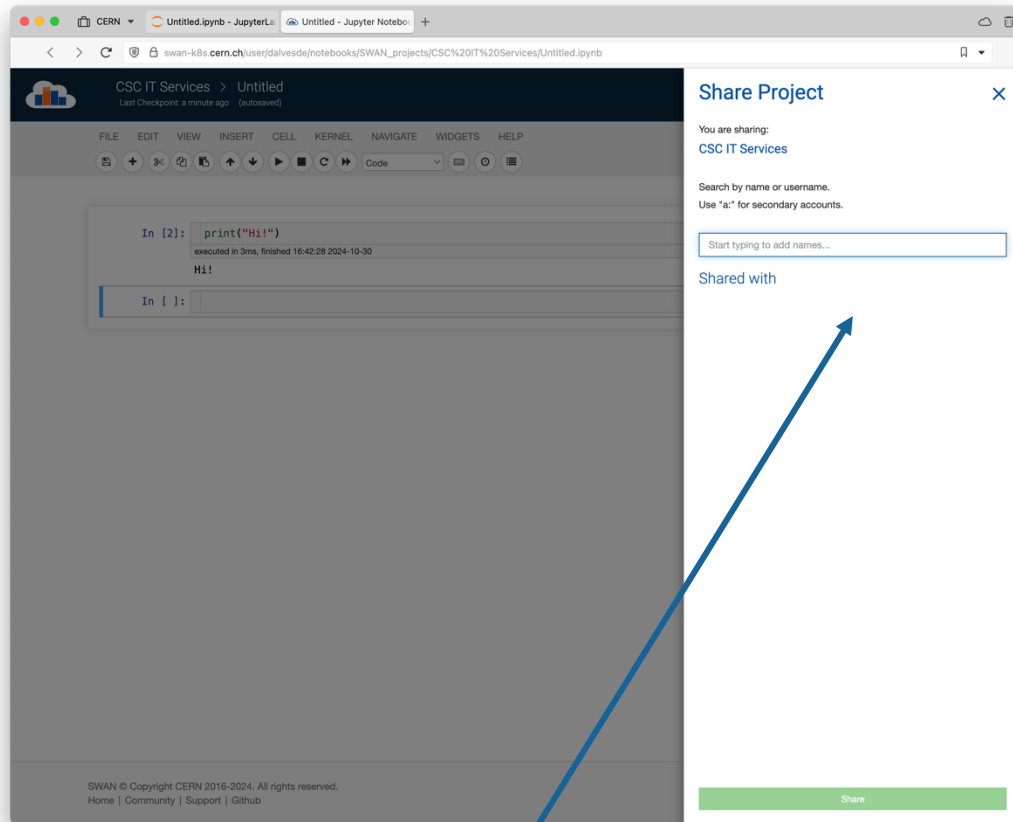
The image shows a JupyterLab interface with a file browser on the left and a code editor on the right. The code editor contains a single cell with the code `print("Hi!")` and its output `Hi!`. A menu is open over the code editor, showing options: New Notebook, Open..., Make a Copy..., Save as..., Rename..., Save and Checkpoint, Revert to Checkpoint, Print Preview, Download as, Trusted Notebook, and Close and Halt. The 'Save and Checkpoint' option is highlighted, and a sub-menu is visible showing two checkpoints: 'Thursday, March 5, 2020 2:26 PM' and 'Thursday, March 5, 2020 2:31 PM'. A blue callout box with the text 'Force save + create version (checkpoint)' has arrows pointing to the 'Save and Checkpoint' menu item and the file browser.

Force save + create version (checkpoint)

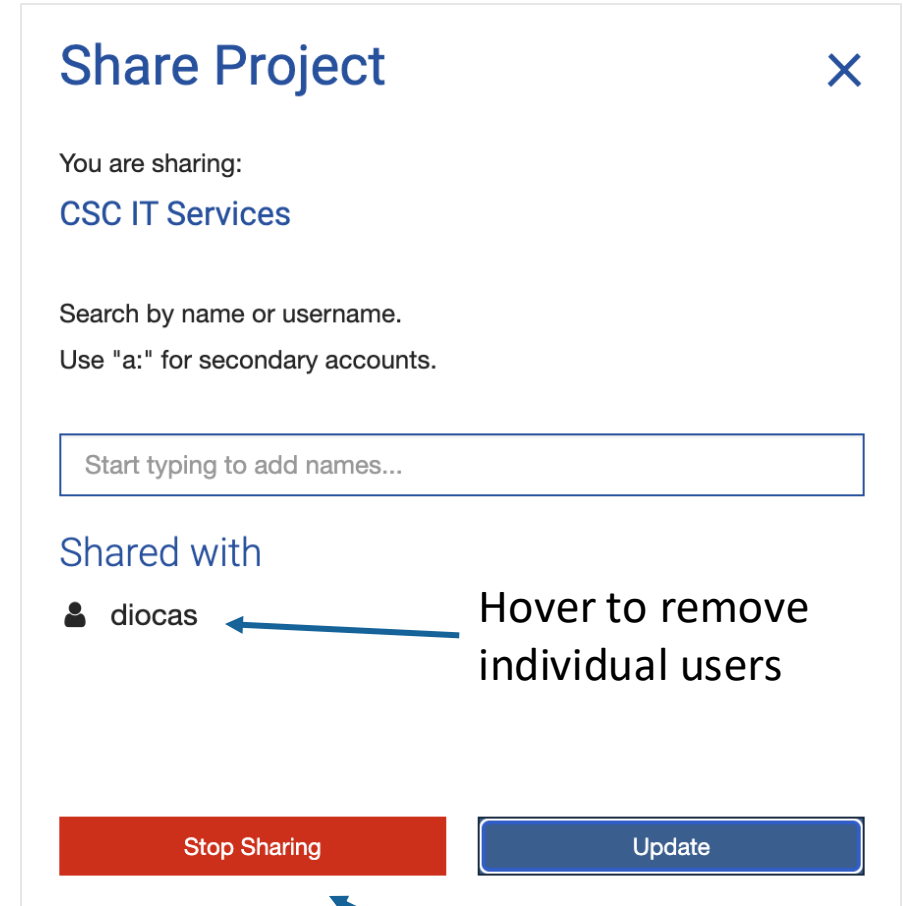




Sharing (Classic UI)



Search like from CERNBox ("a:" for secondary account)
(Egroups support released early next year)



Remove the share
for everyone



GPUs & Machine Learning



GPUs in SWAN

- › SWAN allows attaching a GPU to a user session
 - We currently offer 18x Tesla T4 + 4x A100
- › The GPUs are used **interactively**
 - When starting their session, the user selects a CUDA software stack and gets a GPU
 - GPU-enabled packages can then be used in a notebook and computations offloaded to the GPU by default

```
In [1]: import tensorflow as tf

tf.debugging.set_log_device_placement(True)

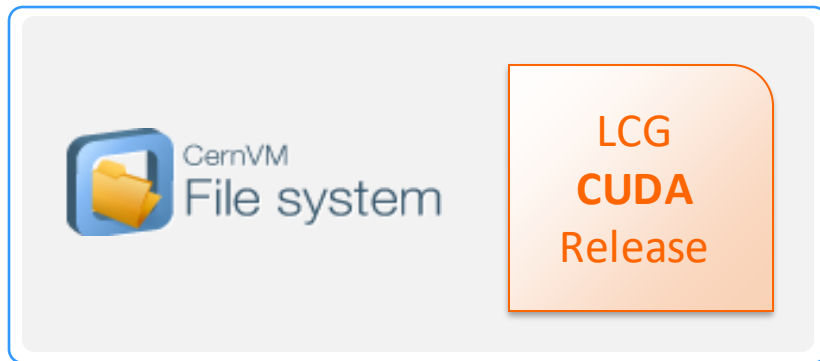
# Create some tensors
a = tf.constant([[1.0, 2.0, 3.0], [4.0, 5.0, 6.0]])
b = tf.constant([[1.0, 2.0], [3.0, 4.0], [5.0, 6.0]])
c = tf.matmul(a, b)
```

```
Executing op MatMul in device /job:localhost/replica:0/task:0/device:GPU:0
```



ML software on CVMFS

- › Software provisioning for ML applications via CVMFS
 - **LCG CUDA** stacks with GPU-enabled software for ML



Apache Spark



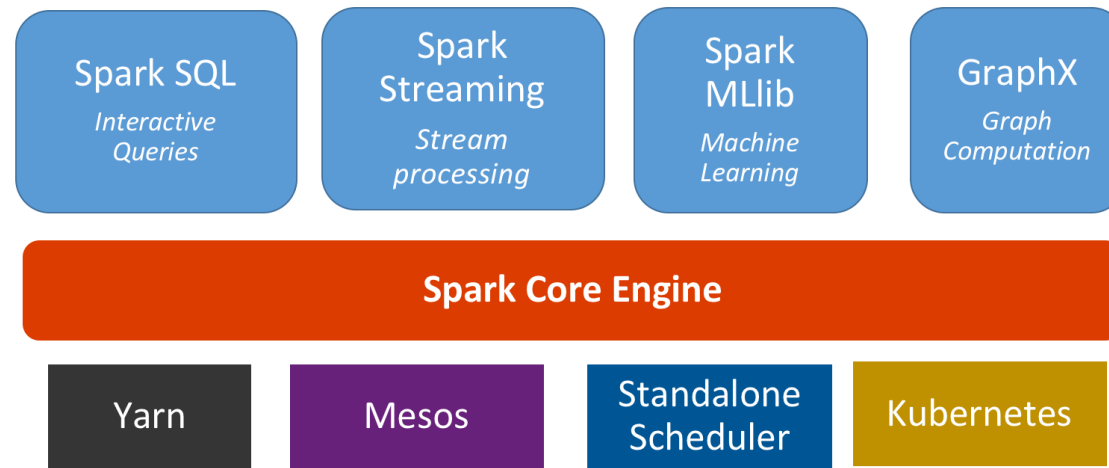


What is Apache Spark?

> Apache Spark

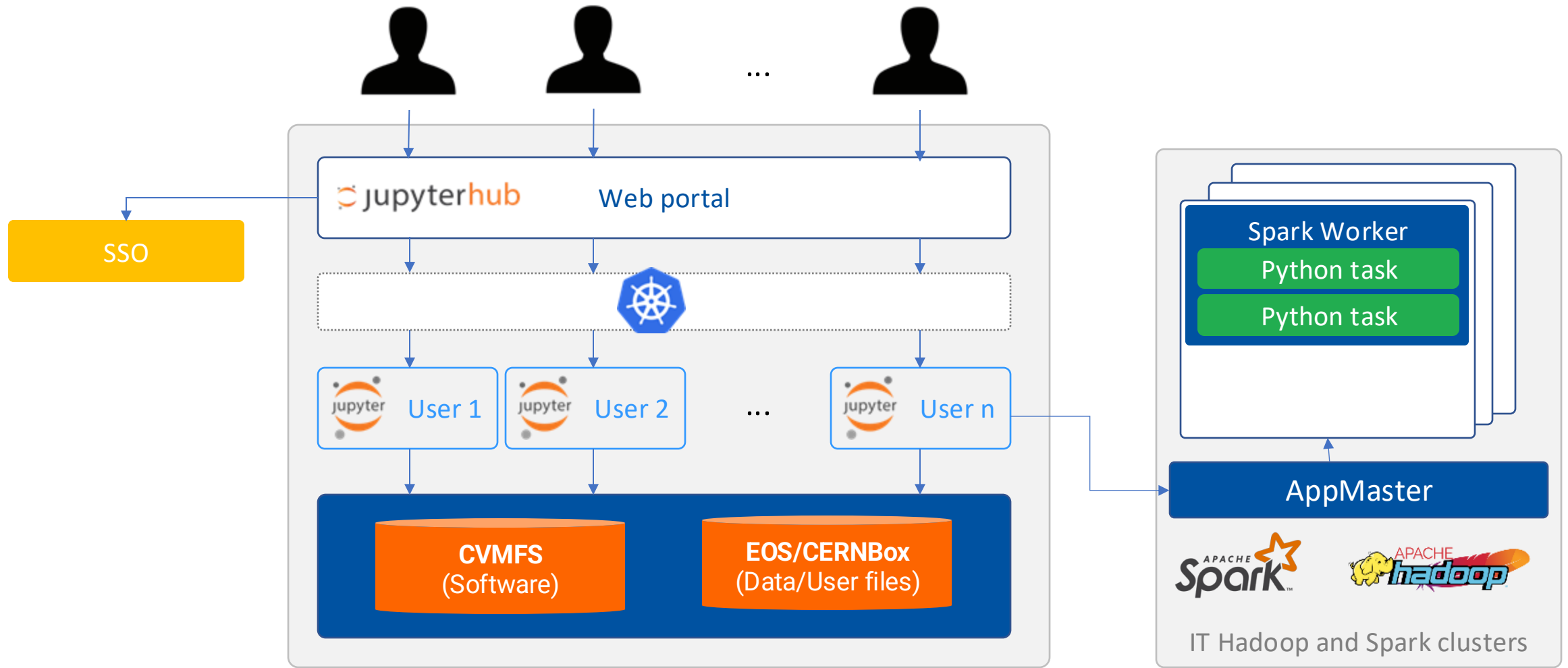
- An open-source parallel processing framework with expressive development APIs (in multiple languages) that allows for sophisticated analytics, real-time streaming and machine learning on large datasets

> Spark ecosystem





SWAN and Spark Architecture





Spark Clusters

Cluster Name	Configuration	Primary Usage
analytix	48 nodes (Cores – 3456, Mem – 34.49TB, Storage – 19.84 PB)	General Purpose
nxcals	49 nodes (Cores – 2712, Mem – 24.34TB, Storage – 16.76 PB)	Accelerator logging (NXCALS) project dedicated cluster
Cloud containers	16 nodes (Cores 256, Mem – 1.87 TB, Storage – EOS)	General Purpose Compute ONLY

Configure Environment ✕

Specify the parameters that will be used to contextualise the container which is created for you. See [SWAN service website](#) for more details and contact to administrators.

[Try out our new experimental interface based on JupyterLab and let us know your feedback!](#)

User Interface [more...](#)

Try the new JupyterLab interface (experimental)

Software stack [more...](#)

105a

Use Python packages installed on CERNBox

Platform [more...](#)

AlmaLinux 9 (gcc13)

Environment script [more...](#)

e.g. \$CERNBOX_HOME/MySWAN/myscript.sh

Number of cores [more...](#)

2

Memory [more...](#)

8 GB

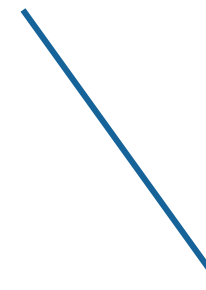
External computing resources

Spark cluster [more...](#)

None

HTCondor pool [more...](#)

None





Spark Connector

The screenshot shows a Jupyter Notebook titled 'Spark_Simple' with a configuration dialog box open on the right. The notebook content includes:

1 Simple example with Spark

This notebook illustrates the use of `Spark` in `SWAN`.

The current setup allows to execute `PySpark` operations on a local standalone Spark instance. This can be done by using the `spark-submit` command.

In the future, `SWAN` users will be able to attach external Spark clusters to their notebooks, so they can take advantage of the Spark ecosystem. In this case, the `kernel` will be added to use Spark from Scala as well.

1.1 Import the necessary modules

The `pyspark` module is available to perform the necessary imports.

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

1.2 Create a SparkContext

A `SparkContext` needs to be created before running any Spark operation. This context is linked to the `SparkSession`.

```
In [ ]: # swan_spark_conf.toDebugString()
```

```
In [ ]: # swan_spark_conf.set('spark.executor.extraJavaOptions', '-Dlog4j.configuration=')
```

- > **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





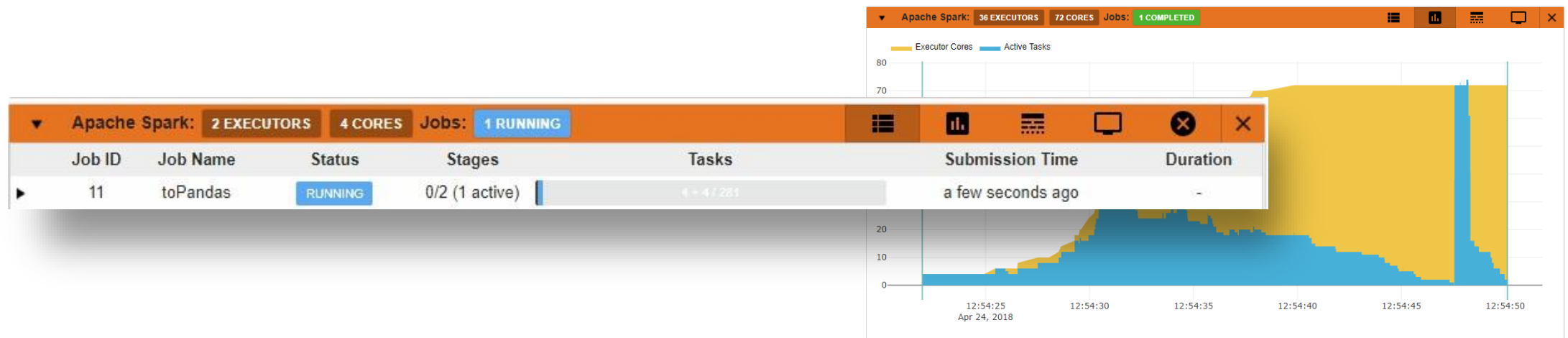
Spark Monitor

> Spark Monitor – Jupyter notebook extension

- For live monitoring of spark jobs spawned from the notebook
- A graph showing number of active tasks & executor cores vs time
- A timeline which shows jobs, stages, and tasks



Google Summer of Code



How to get started?



How to get started?

- > A possible way to start using SWAN is by accessing some content (notebooks) that some other user created
- > In order for a user to share content with others, SWAN offers several options:
 - For exploratory work, collaborative editing: use **sharing** feature (currently only available in the classic UI)!
 - To publish something that can be helpful to a broader audience: use **galleries** (<https://swan-gallery.web.cern.ch/>)!
 - For an event (training/course), for attendees to access its content:





The galleries

- > Galleries of sample notebooks for varied usage of SWAN
 - Quick way to be productive
 - Also accessible from <https://swan-gallery.web.cern.ch/>

Gallery

Basic Examples

ROOT Primer

Accelerator Complex

Beam Dynamics

Machine Learning

Apache Spark

Outreach

AWAKE

Basic Examples

This is a gallery of basic example notebooks: click on the images to inspect the underlying document, open in SWAN the single notebooks or the full git repository!

Open in SWAN

Many of the notebooks are ROOTbooks, based on the ROOT framework. To know more about ROOT, visit root.cern.ch.

Simple ROOTbook (Python)

Simple ROOTbook (C++)

Simple Fitting

Future work



Future work

- › Set **JupyterLab** as the default user interface
- › Allow users to create **custom software environments** (i.e. independent of LCG stacks)
 - E.g. conda
- › Support **experiment software stacks** for analysis
 - E.g. LHCb analysis environment
- › Allow selection of **specific GPU models** when starting a session
- › Deploy new **SWAN instance for ATS**
 - Exposed to devices in the Technical Network

Getting help & contact



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
 - Request 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 features of the tool

Help

1. Introduction

- > What is SWAN
- > Jupyter notebooks
- > Cloud storage: CERNBox and EOS
- > Software: CVMFS

2. Create and manage a SWAN session

- > Select a configuration
- > Set a configuration as default
- > Switch to a new configuration
- > Terminate a session

3. Working with SWAN

- > Create a Project
- > Create a Notebook
- > Create a Folder
- > Open a Terminal



Where to find us

> Contacts

- swan-contact@cern.ch
- <http://cern.ch/swan>

> Code repository

- <https://github.com/swan-cern>

Introduction to SWAN

Thank you

Diogo Castro (diogo.castro@cern.ch)

Enric Tejedor (etejedor@cern.ch)

Pedro Maximino (pedro.maximino@cern.ch)