Evolving SWAN towards an Analysis Facility system

Diogo Castro

On behalf of the SWAN team

https://cern.ch/swan

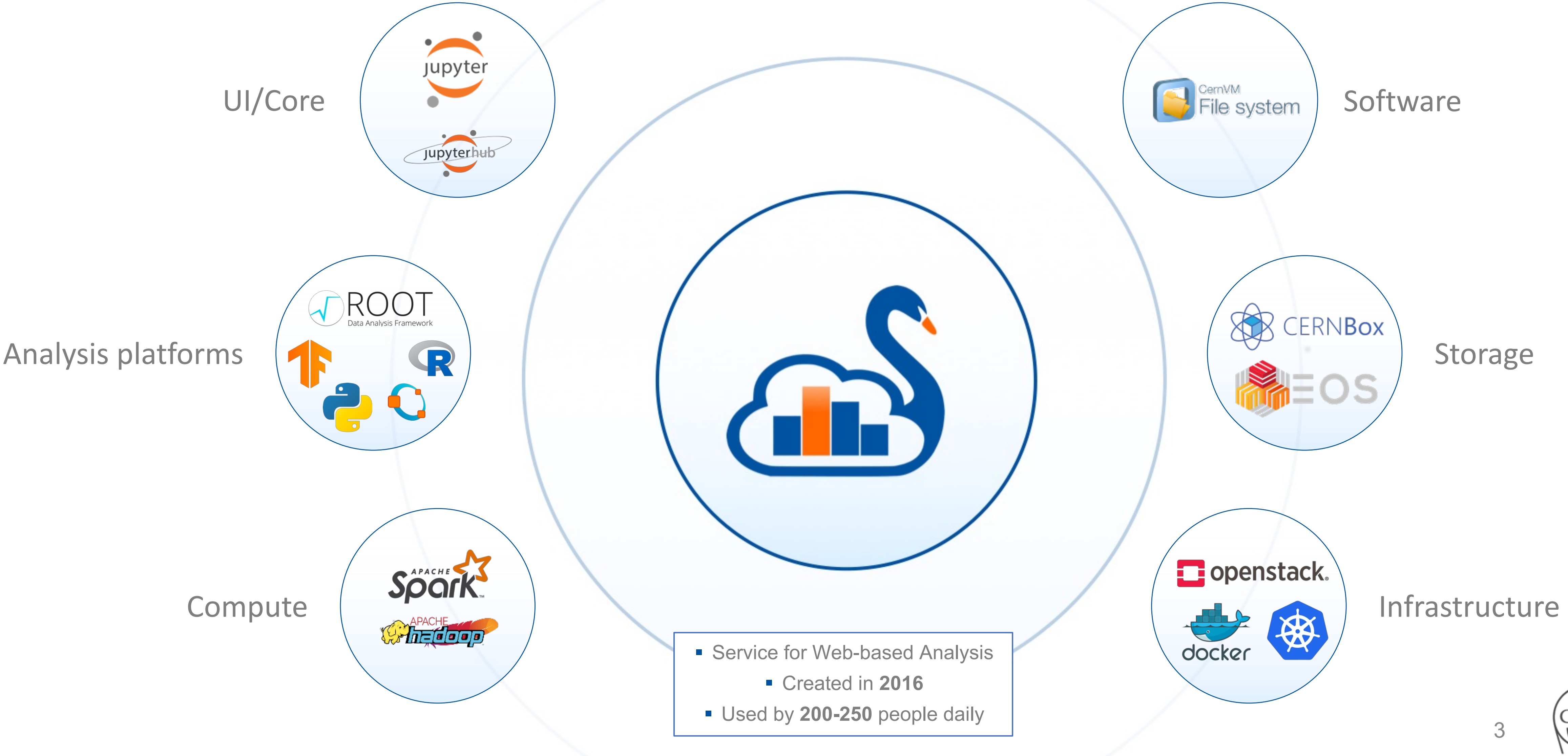
March 6th, 2023

CS3 2023 - Cloud Storage Synchronization and Sharing

A reminder on SWAN

And its current status

Integrating (CERN) services



UI/Core

Analysis platforms

Compute

Software

Storage

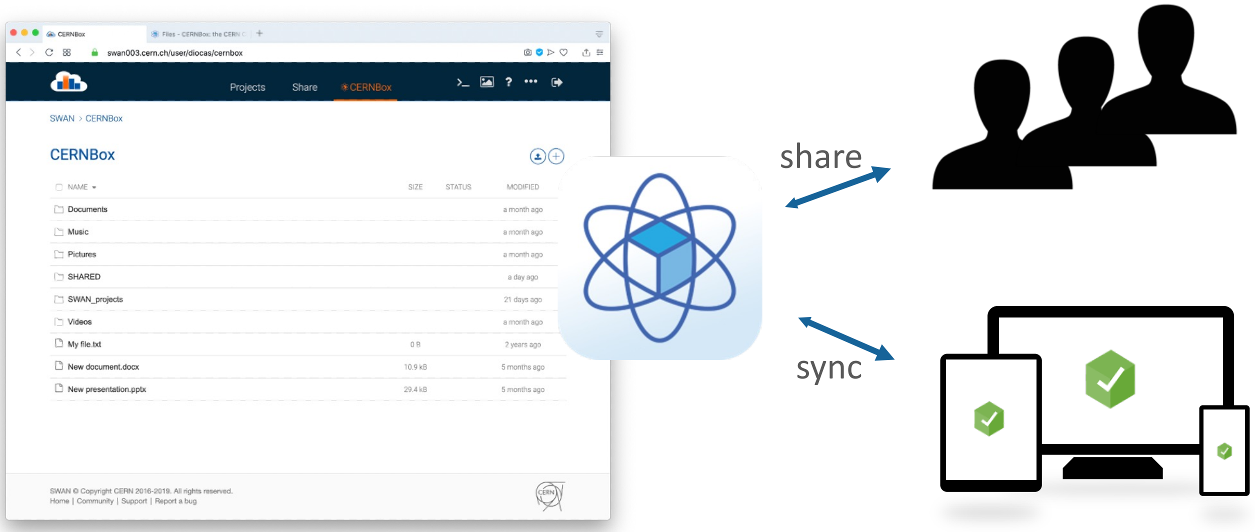
Infrastructure

- Service for Web-based Analysis
- Created in **2016**
- Used by **200-250** people daily

Storage

- > All the data our users need for their analysis
  - CERNBox as home directory
  - Experiment repositories, projects, open data, …
  - (EOS Fuse client)

- > Sync&Share
  - Files synced across devices and the Cloud
  - Simple collaborative analysis
  - (Custom share API)



CERNBox
Files - CERNBox: the CERN C
swan003.cern.ch/user/diocas/cernbox
Projects
Share
CERNBox
share
sync

SWAN > CERNBox

CERNBox

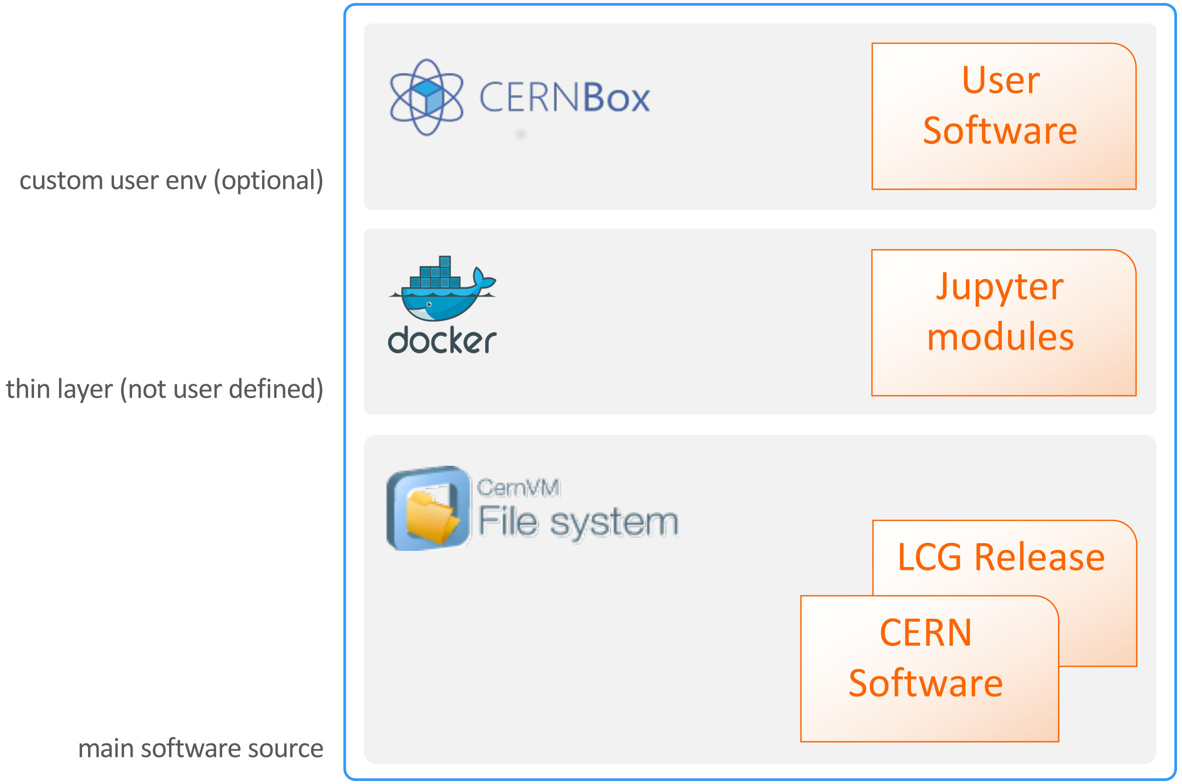
NAME	SIZE	STATUS	MODIFIED
Documents			a month ago
Music			a month ago
Pictures			a month ago
SHARED			a day ago
SWAN_projects			21 days ago
Videos			a month ago
My file.txt	0 B		2 years ago
New document.docx	10.9 kB		5 months ago
New presentation.pptx	29.4 kB		5 months ago

SWAN © Copyright CERN 2016-2019. All rights reserved.
Home | Community | Support | Report a bug

Software

> Software distributed through CVMFS
- ”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



custom user env (optional)

CERNBox

User Software

thin layer (not user defined)

docker

Jupyter modules

main software source

CernVM File system

LCG Release

CERN Software

Current project priorities

1
- Conclude migration to Kubernetes
- Ensure scalability

2
- Conclude migration to Jupyterlab

3
- Migration to Alma 9 / simplification of current docker images
- Update to latest versions of upstream

4
- Conclude integration of more CERN services

5
- New ways to manage software
- Binder

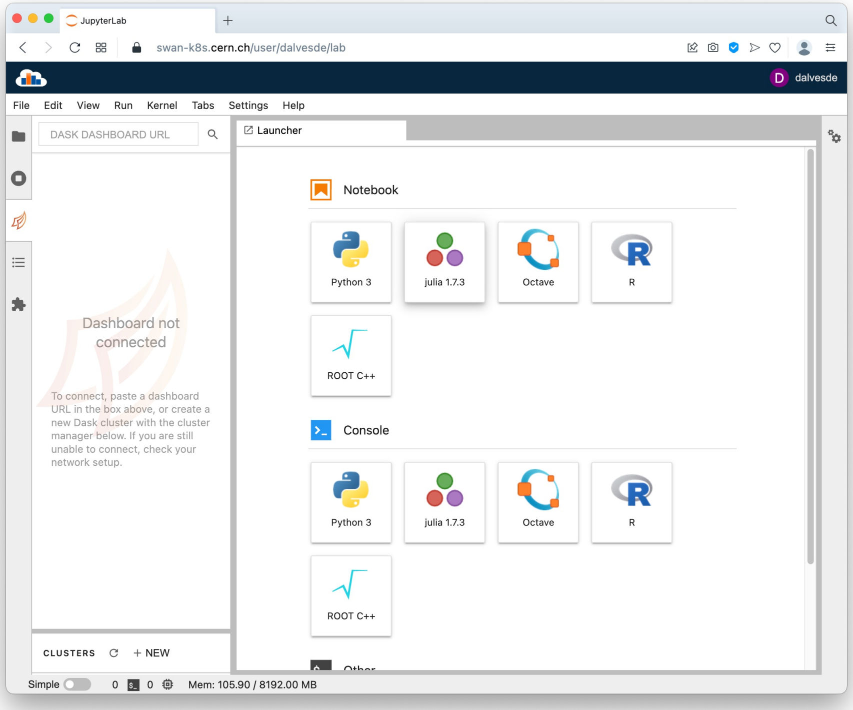
Migration to Lab

> Final stages of migration

- Spark (monitor, connector) extensions migrated
- Other small extensions+branding, also done
- Looking into other integrations (i.e Dask)

> Deeper Sync&Share integration

- Ongoing integration with CERNBox using the CS3 APIs Jupyterlab extension [1]
- Full sharing and collaborative capabilities
- Easier integration of SWAN with other CS3 enabled EFSS



Dashboard not connected

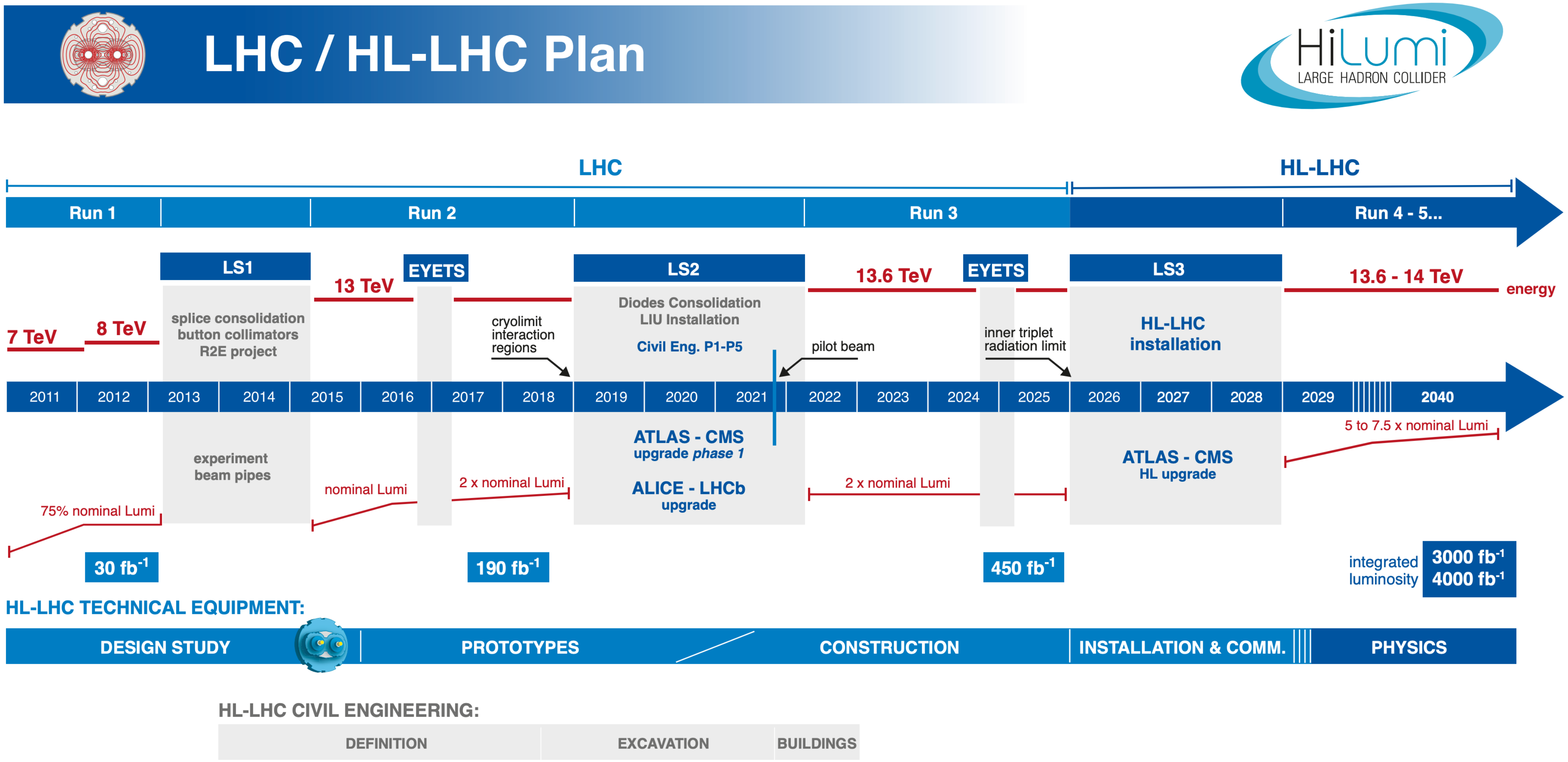
To connect, paste a dashboard URL in the box above, or create a new Dask cluster with the cluster manager below. If you are still unable to connect, check your network setup.

[1] See "Science Mesh demos" (Cs3api4lab), Tue @ 14:50

Why an analysis facility?

And why SWAN needs to evolve

The road to Hi-Lumi



LHC / HL-LHC Plan

HiLumi LARGE HADRON COLLIDER

LHC

HL-LHC

Run 1 | Run 2 | Run 3 | Run 4 - 5...

LS1: splice consolidation, button collimators, R2E project; experiment beam pipes

EYETS

LS2: Diodes Consolidation, LIU Installation, Civil Eng. P1-P5; ATLAS - CMS upgrade *phase 1*; ALICE - LHCb upgrade

EYETS

LS3: HL-LHC installation; ATLAS - CMS HL upgrade

energy: 7 TeV, 8 TeV, 13 TeV, 13.6 TeV, 13.6 - 14 TeV

cryolimit interaction regions

pilot beam

inner triplet radiation limit

2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2040

75% nominal Lumi

nominal Lumi

2 x nominal Lumi

2 x nominal Lumi

5 to 7.5 x nominal Lumi

30 fb^{-1}

190 fb^{-1}

450 fb^{-1}

integrated luminosity: 3000 fb^{-1}, 4000 fb^{-1}

HL-LHC TECHNICAL EQUIPMENT:

DESIGN STUDY | PROTOTYPES | CONSTRUCTION | INSTALLATION & COMM. | PHYSICS

HL-LHC CIVIL ENGINEERING:

DEFINITION | EXCAVATION | BUILDINGS

Why an analysis facility?

- HL-LHC needs are pushing us to build modern Analysis Facilities
  - Traditional batch processing
  - Interactive computing on big datasets, with new interfaces (Jupyter)
- An AF should facilitate access to:
  - Software
  - Storage (+ sharing)
  - Computing resources (elastic)
- Ongoing effort to provide an AF @ CERN
  - Interdepartmental collaboration

Computing

2x Servers	10x Compute power

Storage

1.5EB Disk	2.5EB Tape

Build on what already exists

> Evolution as opposed to a new and dedicated AF

- Less costs and quicker to deploy
- SWAN is a good candidate for the interface of an AF @ CERN!

> SWAN needs to be an entry point to external and heterogeneous resources

- Multiple services already available at CERN
- We don’t want to run the full infrastructure
- More freedom to users to chose what best fits their use cases

Resources integration

Past, ongoing and future work

Integration model

> A Platform for physics analysis

- With support for both *single-node* and *distributed* analysis
- Keep only lightweight resources “local”

> SWAN acts as a client to other resources

- Allows connection to multiple types of resources, instead of creating a session on them directly
- Doesn’t sacrifice the usablity (users have access to the system without delay)
- Test small and local (quick) and run big and distributed
- Keeps the independence between services (i.e upgrade schedules, dependencies, etc)

Spark

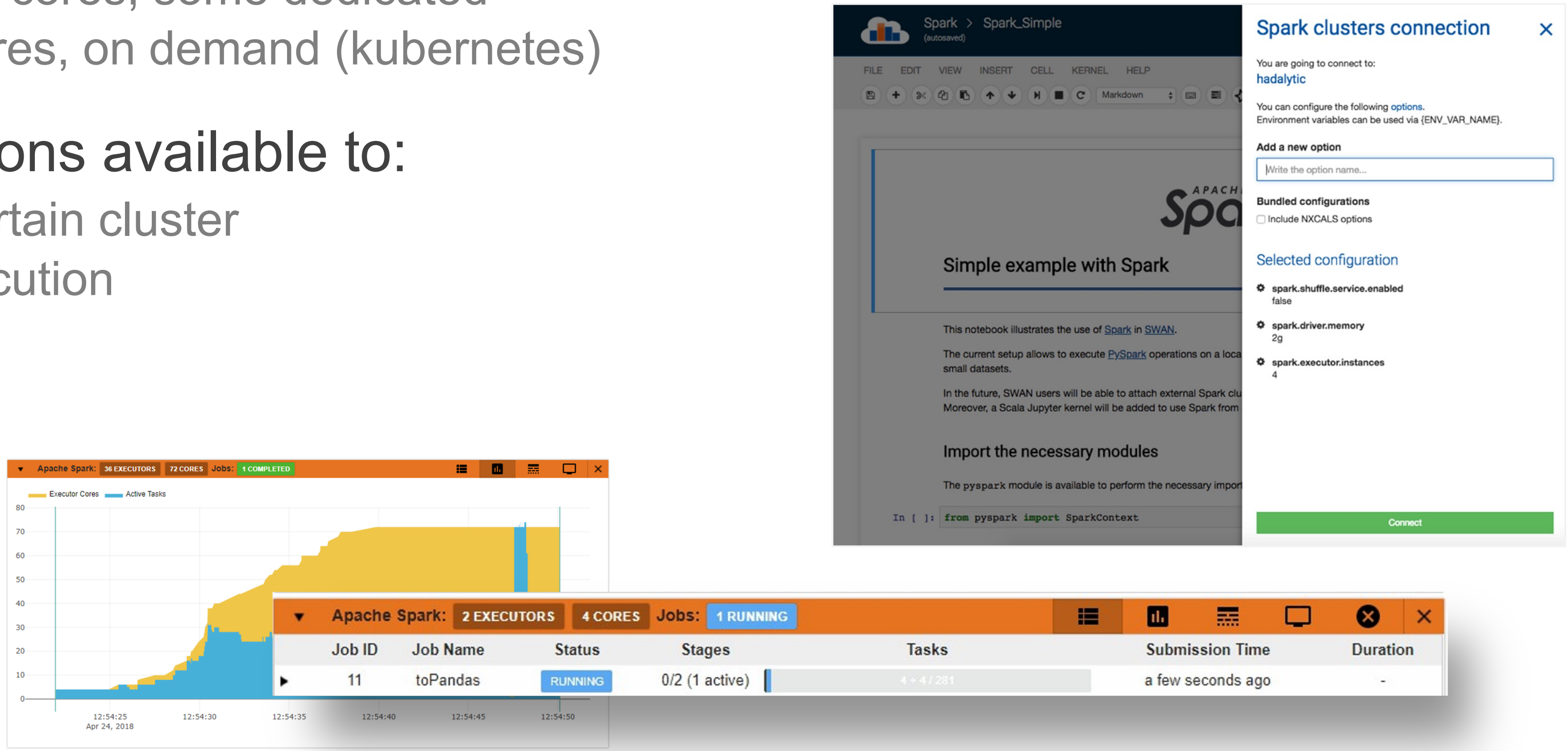
Production

> SWAN is connected to the Spark clusters at CERN

- Physical: ~3800 cores, some dedicated
- Virtual: ~250 cores, on demand (kubernetes)

> Jupyter extensions available to:

- Connect to a certain cluster
- Monitor the execution



Simple example with Spark

This notebook illustrates the use of Spark in SWAN.

The current setup allows to execute PySpark operations on a loca
small datasets.

In the future, SWAN users will be able to attach external Spark clu
Moreover, a Scala Jupyter kernel will be added to use Spark from

Import the necessary modules

The pyspark module is available to perform the necessary impor

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

Spark clusters connection ✕

You are going to connect to:
hadalytic

You can configure the following options.
Environment variables can be used via {ENV_VAR_NAME}.

Add a new option

Write the option name...

Bundled configurations

□ Include NXCALS options

Selected configuration

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

Connect

▼ Apache Spark: 2 EXECUTORS 4 CORES Jobs: 1 RUNNING

	Job ID	Job Name	Status	Stages	Tasks	Submission Time	Duration
▶	11	toPandas	RUNNING	0/2 (1 active)	4 + 4 / 281	a few seconds ago	-

GPUs

Production

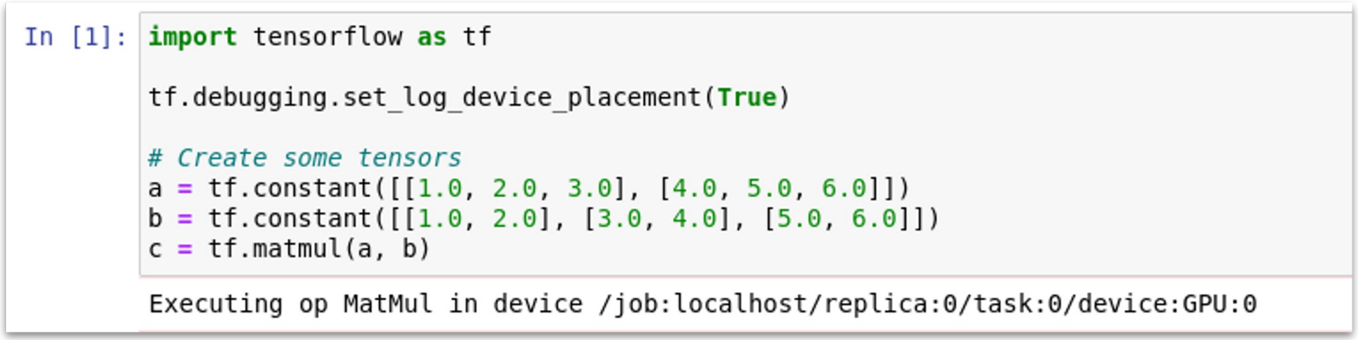
> SWAN allows to attach a GPU to a user session

- Feature of the new SWAN k8s deployment
- ~12 GPUS (Tesla T4)

> The GPUs are used interactively

- When starting their session, the user selects a CUDA software stack and gets a GPU
- GPU-enabled packages (e.g. tensorflow, PyTorch) can then be used in a notebook and offload to the GPU by default

> Currently looking into GPU concurrency



```
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
```

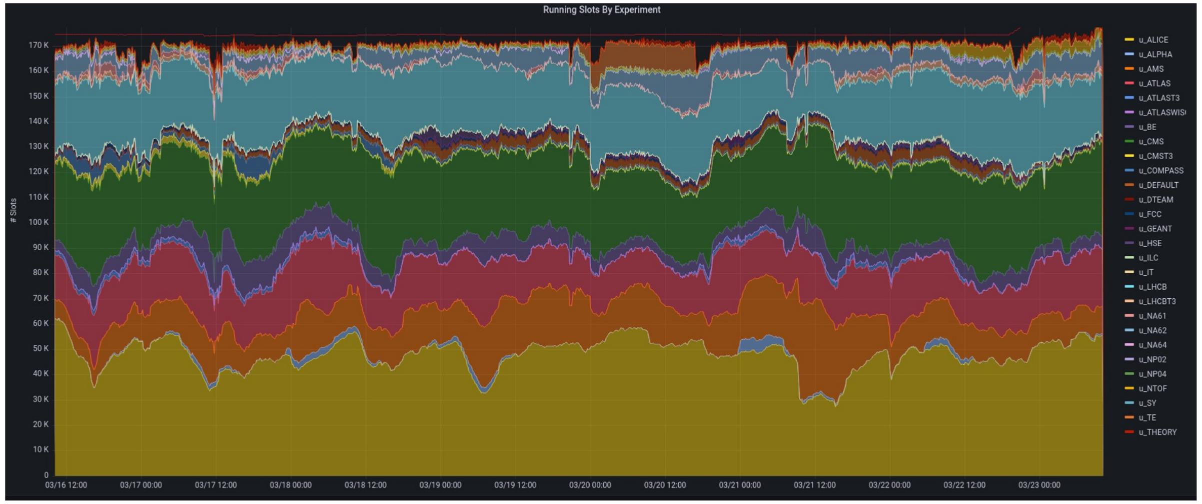
HTCondor

Testing

> Goal: leverage HTCondor pools at CERN from SWAN
  - Up to ~175k cores in shared pools at CERN – limited by the quotas assigned depending on experiment affiliation
  - Already used for analysis

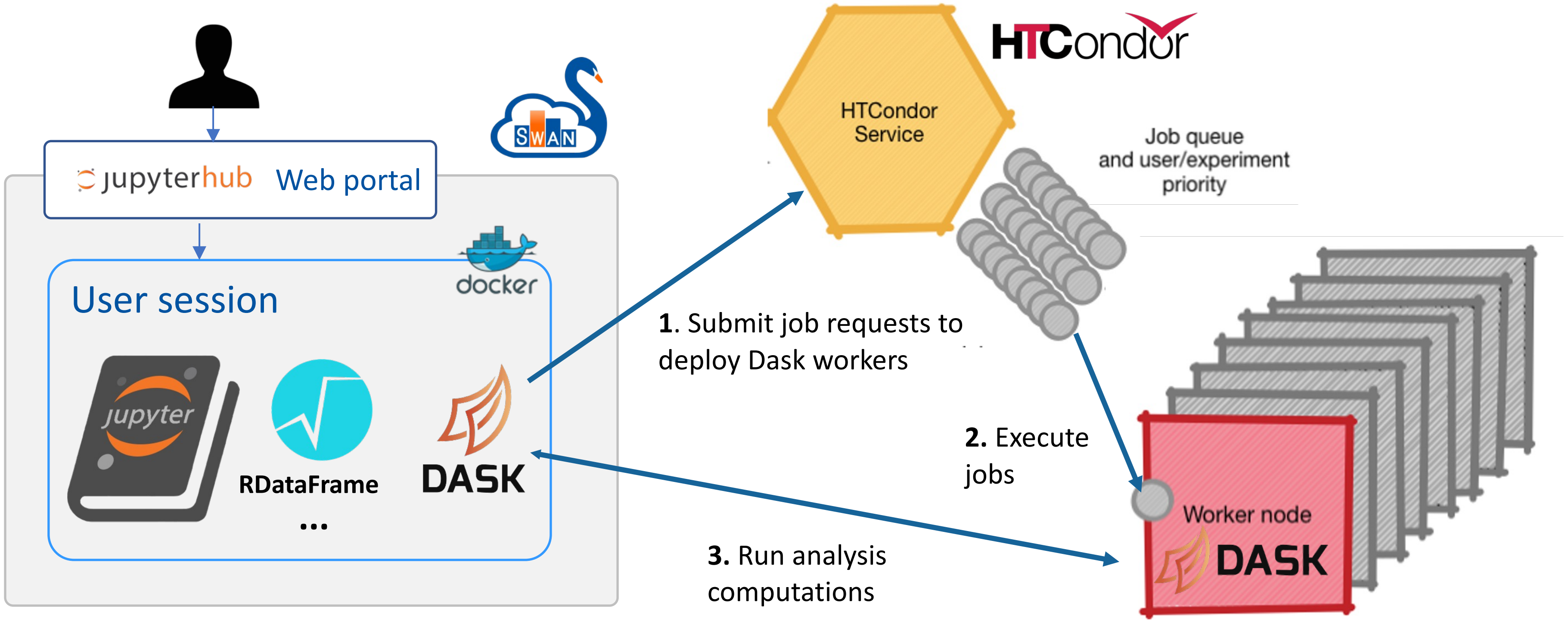
> Batch submission: already supported
  - Condor packages available on CVMFS

> Interactive usage: in pilot phase
  - Collaboration with Batch Service@CERN
  - Dask packages available on CVMFS
  - Will be exposed to users when migration to JupyterLab is finished (Q2 2023)



HTCondor + Dask = interactivity

Final stages of dev



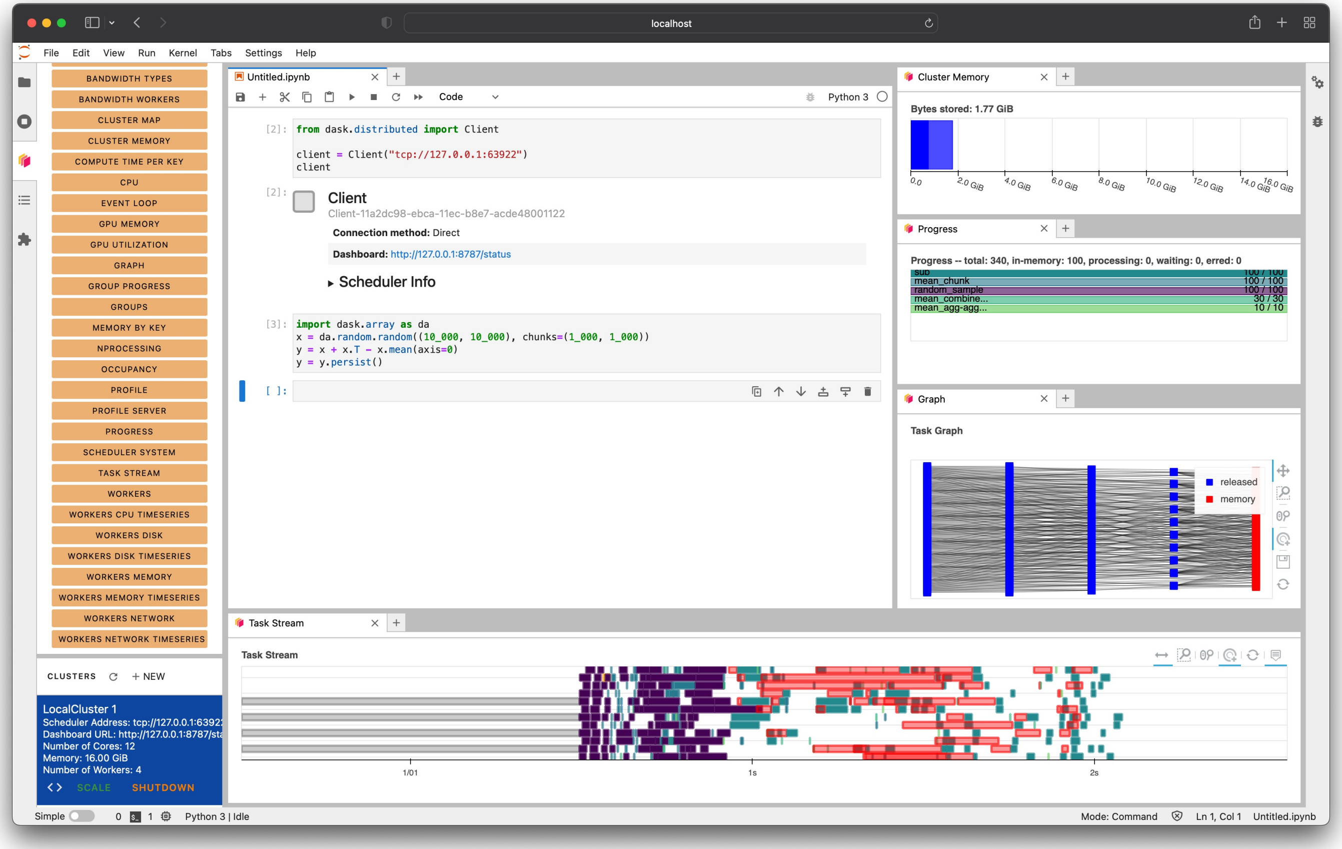
1. Submit job requests to deploy Dask workers

2. Execute jobs

3. Run analysis computations

HTCondor + Dask = interactivity

- Upstream Dask extension
  - A cluster shared across multiple “clients” (notebooks)
  - Better resource utilization
  - Spark connection is evolving into this model as well



```
[2]: from dask.distributed import Client

     client = Client("tcp://127.0.0.1:63922")
     client
```

[2]: **Client**

Client-11a2dc98-ebca-11ec-b8e7-acde48001122

Connection method: Direct

Dashboard: http://127.0.0.1:8787/status

▸ Scheduler Info

```
[3]: import dask.array as da
     x = da.random.random((10_000, 10_000), chunks=(1_000, 1_000))
     y = x + x.T - x.mean(axis=0)
     y = y.persist()
```

```
[ ]:
```

LocalCluster 1
Scheduler Address: tcp://127.0.0.1:6392
Dashboard URL: http://127.0.0.1:8787/sta
Number of Cores: 12
Memory: 16.00 GiB
Number of Workers: 4

< > SCALE SHUTDOWN

HPC

Early dev

> HPC service at CERN

- Applications and use cases that do not fit the standard batch HTC model, typically parallel MPI applications.
  - Ex: Computation Fluid Dynamics, Beam simulation, plasma simulation, …)
- Uses Ceph FS between submission and work nodes

> SWAN integration ongoing

- Testing authentication and software requirements
- Missing storage integration: CERNBox/REVA/CEPH integration?
  - We don't want shared secrets from other services

> Reva Ceph FS "simplified" storage backend

- PoC available, some features (sharing/ACLs, snapshotting) not available but not needed

Other future possibilities

> Reana
- CERN’s Reusable Analysis Platform
- Move from exploratory analysis into a reproducible format

> Kubeflow
- Machine learning and MLOps service at CERN
- Create and deploy pipelines directly from SWAN

Conclusion

Takeaways

> The HL-LHC upgrade will bring many challenges

- In terms of storage and computing
- But also on the complexity of the analysis

> The CERN infrastucture is evolving to cope with the load

- And a new Analysis Facility workgroup was formed

> SWAN is a good candidate for the interface of an AF @ CERN

- Gives access to the software, storage and UX expected by the users
- More services need to be integrated

Where to find us

> Contacts

- swan-admins@cern.ch
- http://cern.ch/swan
- https://swan-community.web.cern.ch/

> Repository

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

> Science Box [1]

- (deploys the SWAN Helm Chart)
- https://cern.ch/sciencebox

[1] See “Driving the ScienceBox package into the future”, Wed @ 11:45

Evolving SWAN towards an Analysis Facility system

Thank you

Diogo Castro
diogo.castro@cern.ch