Evolving SWAN through simplification

Diogo Castro

On behalf of the SWAN team

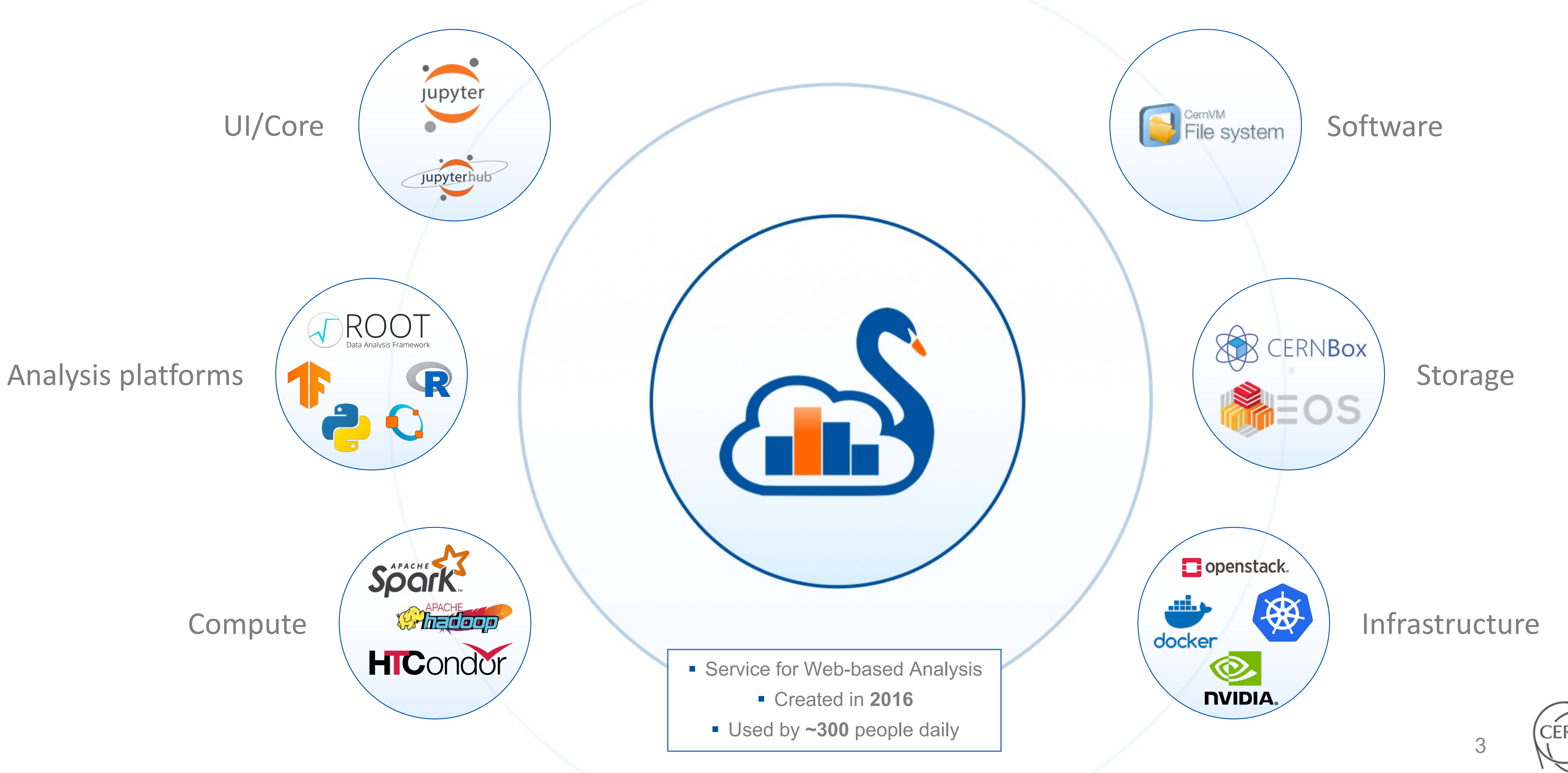
https://cern.ch/swan

March 12th, 2024

CS3 2024 - Cloud Storage Synchronization and Sharing

A reminder on SWAN

Integrating (CERN) services

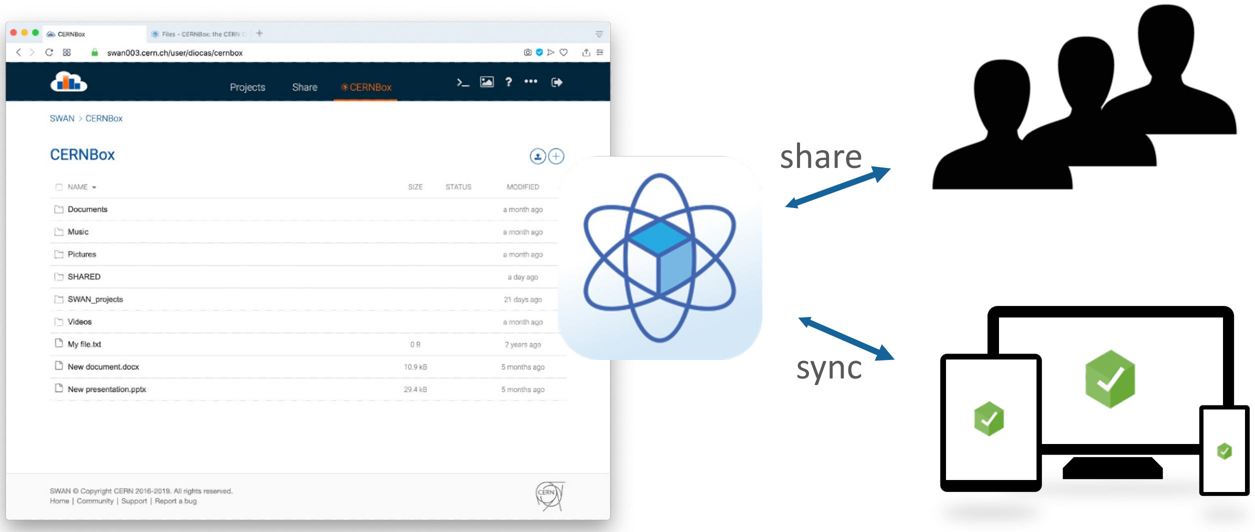


- Service for Web-based Analysis
- Created in **2016**
- Used by **~300** people daily

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


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

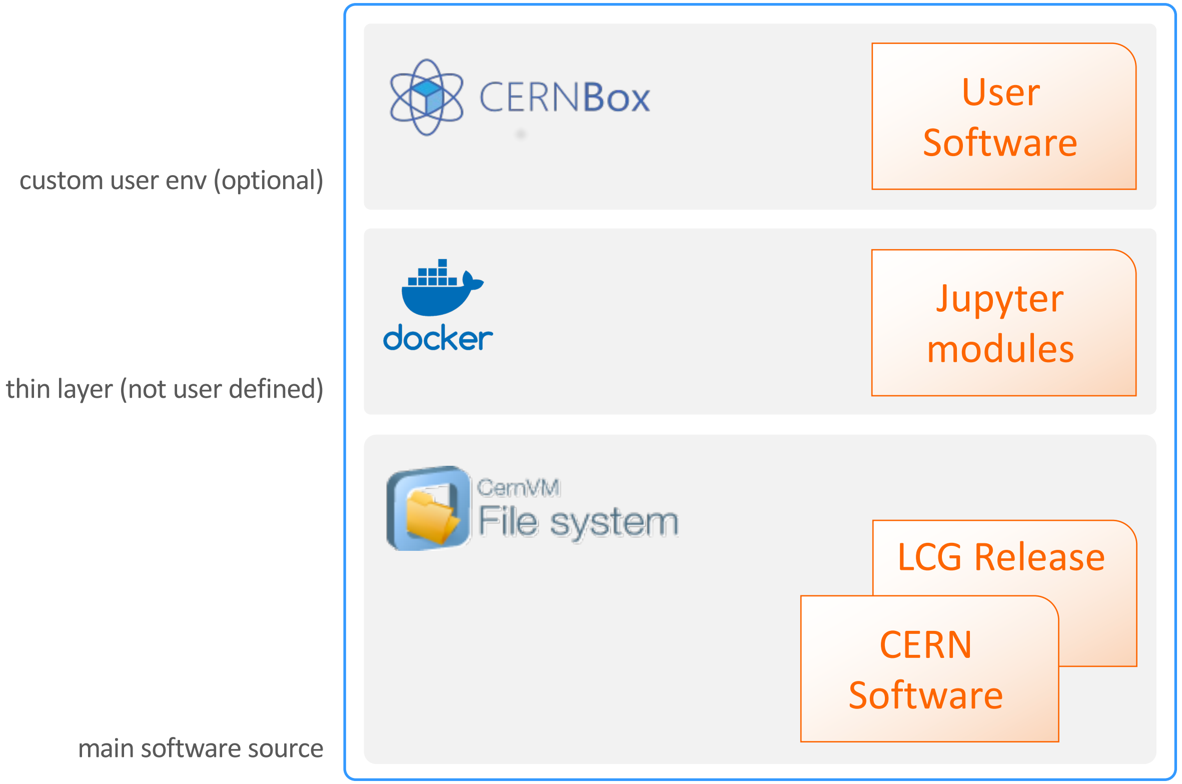
CERNBox

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



Latest updates

Project priorities in 2023

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 Kubernetes

1

> Migration campaign finished on March 5th, 2024

- All physical nodes have been removed from the service
- Improved operations and a single source of truth for metrics
- Better UX for users (aligned set of features, single point of entry)

> Lessons learned on the way

- Some components' maturity had to stabilize
- Need to gain operational expertise
- Physical machines were used as a fallback to disruptive updates

> Further improvements (in progress)

- Blue/Green deployment for cluster (disruptive) updates
- DevOps automated tools

GPU access

1

- SWAN allows to attaching a GPU to a user session
  - Feature of the new SWAN k8s deployment, now available to anyone
  - 18 GPUS (Tesla T4)

- New project: CERN IT-wide resources sharing
  - Sharing all GPUs across different services in a pool
  - Easy scalability in case of need (e.g., for tutorials)

GPU partitioning

1

> A new NVIDIA GPU operator deployed

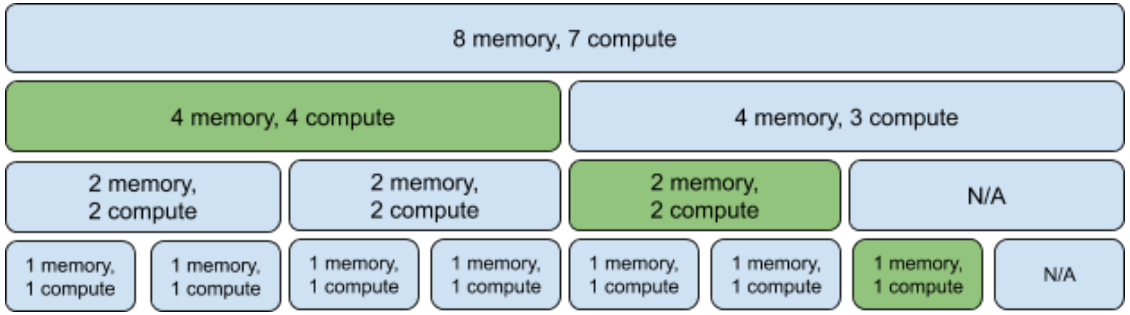
- Supports newer GPU cards, including partitionable GPUs, i.e. Multi-Instance GPUs (MIG is viable with A100s, that are scarce at CERN)

> With GPU partitioning, users have exclusive access to one GPU fragment

- No interferences

> Partitioning GPUs allows for resource sharing and better resource utilization

- E.g. assign one fragment per participant of a tutorial



8 memory, 7 compute

4 memory, 4 compute | 4 memory, 3 compute

2 memory, 2 compute | 2 memory, 2 compute | 2 memory, 2 compute | N/A

1 memory, 1 compute | 1 memory, 1 compute | 1 memory, 1 compute | 1 memory, 1 compute | 1 memory, 1 compute | 1 memory, 1 compute | 1 memory, 1 compute | N/A

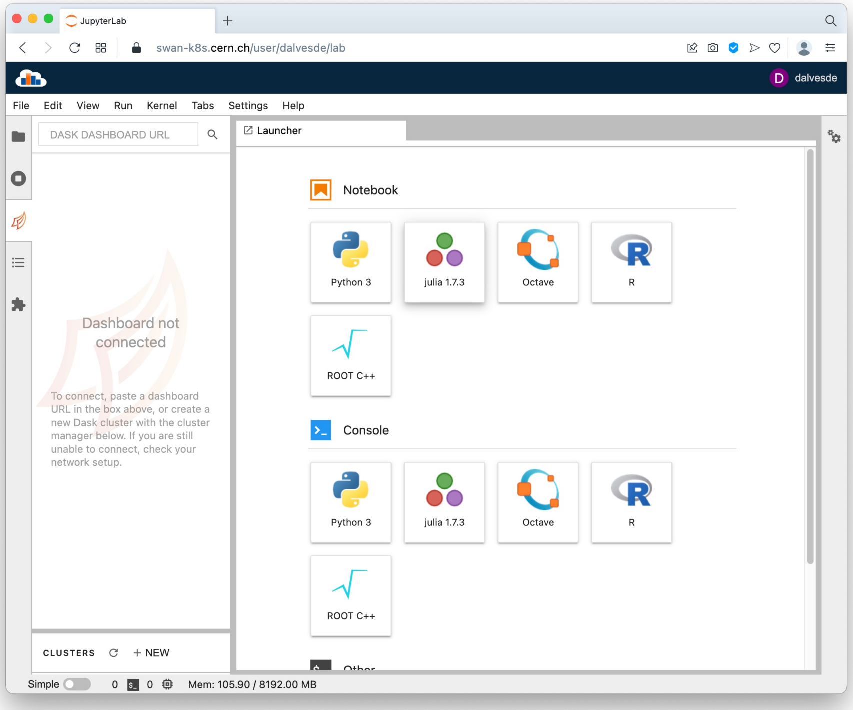
Migration to JupyterLab

2

> Deployed Jupyterlab v4
- Extensions migrated to the new version

> Available as beta UI
- Collection of user feedback underway
- Users can use the old UI in parallel

> Deeper Sync&Share integration
- Ongoing integration with CERNBox using the CS3 APIs Jupyterlab extension (CS3Mesh project)
- Full sharing and collaborative capabilities
- Currently migrating to Lab v4 and making UI production ready



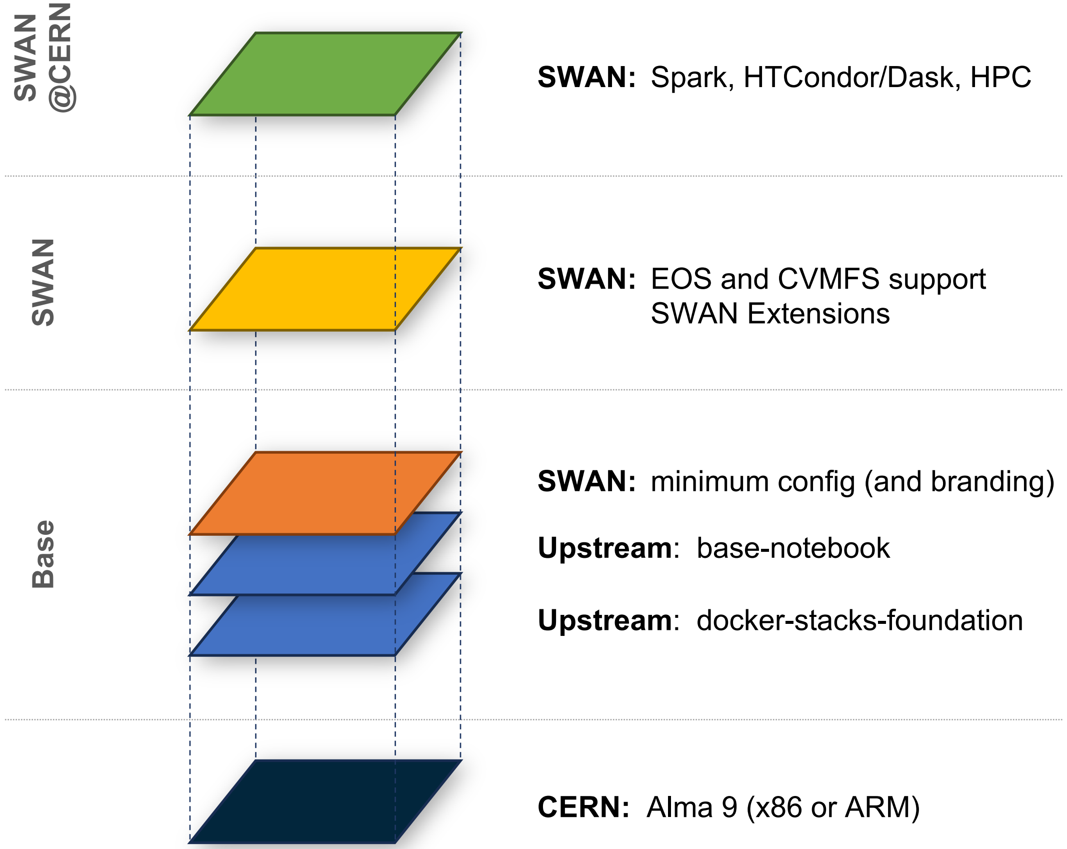
Dashboard not connected

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.

Migration to Alma 9

3

- Key SWAN container images migrated from CERNCentOS7 to Alma9
  - User session image (Jupyter server)
  - JupyterHub image
- User images rewritten from scratch
  - Like upstream images, but on top of Alma 9 instead of Ubuntu
  - Same entry points and configuration options
- Modular components' configuration
  - EOS, CVMFS, External resources, etc, are independently configured on separate scripts
  - Easy to disable or add new components



SWAN: Spark, HTCondor/Dask, HPC

SWAN: EOS and CVMFS support
SWAN Extensions

SWAN: minimum config (and branding)

Upstream: base-notebook

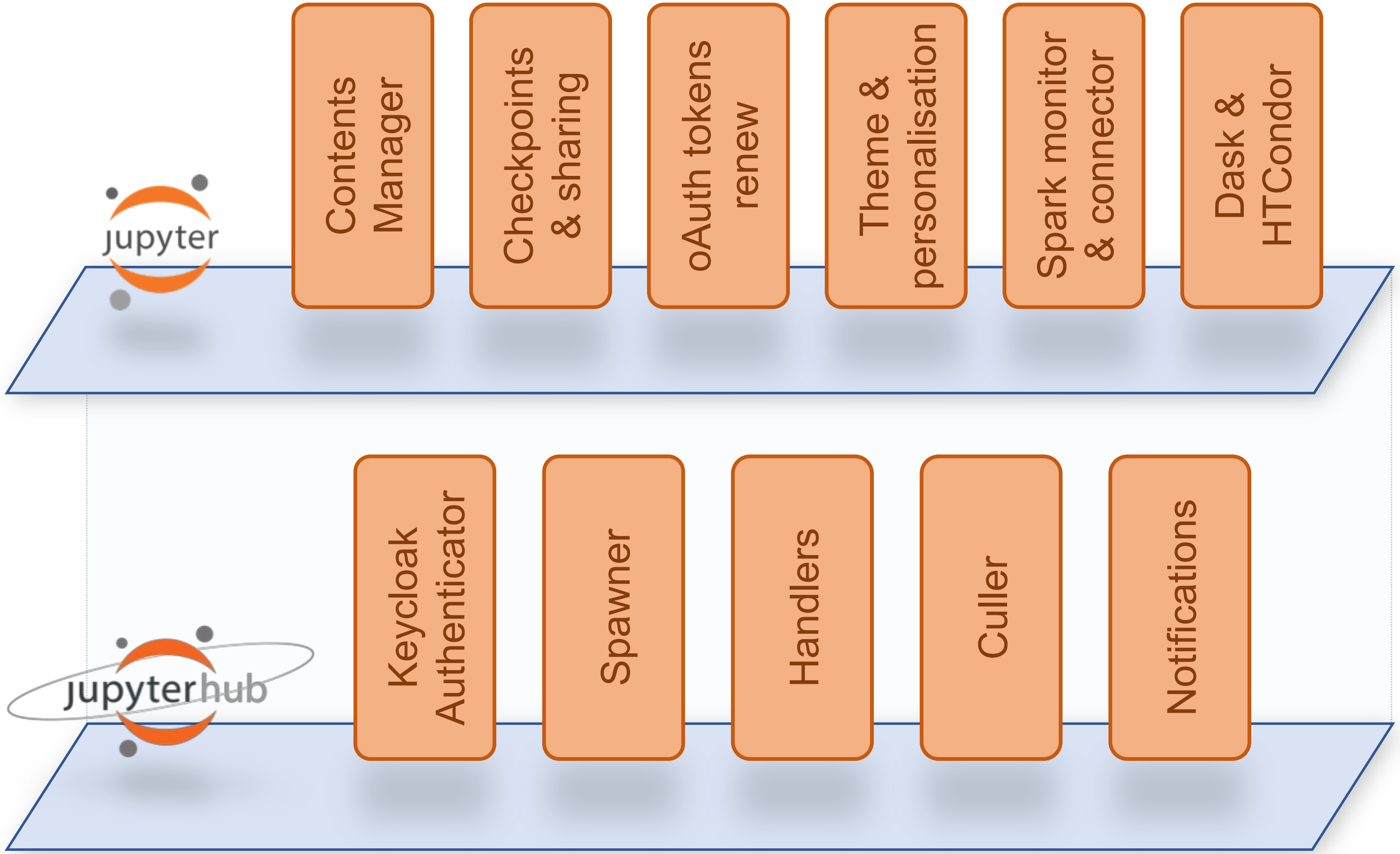
Upstream: docker-stacks-foundation

CERN: Alma 9 (x86 or ARM)

Migration to Alma 9

3

- More runtime freedom
  - They can be run within Jupyterhub (i.e. prod SWAN) but also independently (e.g. locally or headless in a CI)
- All dependencies updated to the latest versions
  - Some SWAN personalizations were replaced with upstream ones



Integration with external resources

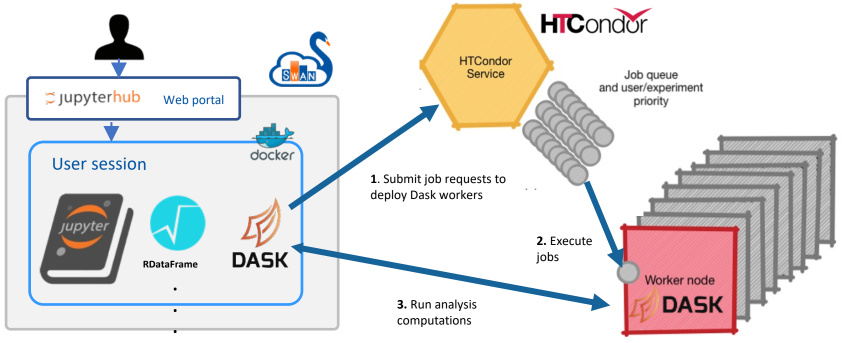
4

- Spark infrastructure is being updated to Alma 9
  - Coordinating with Hadoop service the updates to Alma 9 for Spark on Hadoop
  - Investigating deploying Spark on YARN using container images to streamline the update
- All Spark Jupyter extensions have been updated to Lab 4
- CERN HPC integration is now in QA
  - Applications and use cases that do not fit the standard batch HTC model, typically parallel MPI applications
  - Uses CEPH as shared storage between submission and worker nodes
  - CEPH FS integrated as PVCs in SWAN, mounted only for allowed users

Analysis facility pilot

4

- Support interactive distributed analysis for High Energy Physics
  - Address the future analysis needs due to foreseen increase in data volumes.
- Dask as the connector to batch resources
  - The two main HEP analysis frameworks, ROOT and coffea, rely on Dask for running analysis distributedly
- For now, it uses overcommitted "static" slots on HTCondor
  - Optimizes usage of batch resources
  - A well-stacked batch farm with a good job mix can get to 80% CPU utilization
  - Known analysis jobs potential to stack nicely with other workloads to drive up utilization



Analysis facility pilot

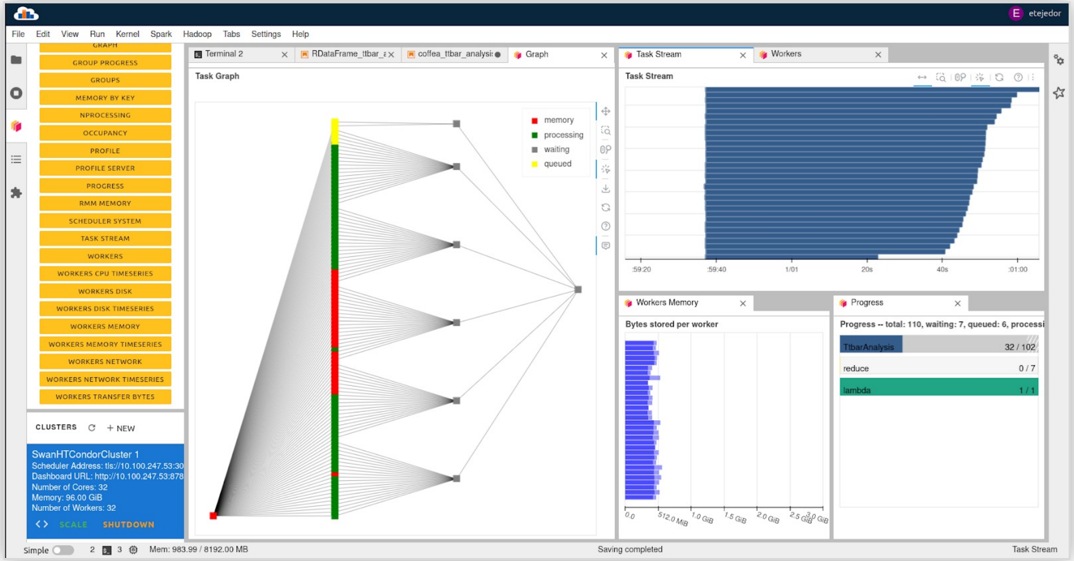
4

> A Pilot has been approved

- Validate demand
- It will validate with real users its usefulness to CERN use cases and the necessity of improvements

> Future improvements

- Use tokens throughout the workflow (currently a Kerberos auth is required)
- Allow users to close the notebook UI and still be able to retrieve the jobs’ status
- Potentially different interactive jobs allocation model
- Improve custom software environment integration



SwanHTCondorCluster 1
Scheduler Address: tls://10.100.247.53:30
Dashboard URL: http://10.100.247.53:878
Number of Cores: 32
Memory: 96.00 GiB
Number of Workers: 32

Progress -- total: 110, waiting: 7, queued: 6, processi

TtbarAnalysis	32 / 102
reduce	0 / 7
lambda	1 / 1

Mem: 983.99 / 8192.00 MB

Saving completed

Virtual environments

5

> Project kickstarted to allow the integration of LHC Control tools in SWAN

> Objective: Make it easier to manage project dependencies and ensure consistent execution across different environments

- So that users can publish projects via Gitlab, and still recreate the environment consistently

> For performance reasons, for now, environments are created locally in the container storage

- We are investigating the feasibility of the EOS Squash FS feature to persist across sessions

> We will try to integrate GUI tools to help add/remove packages

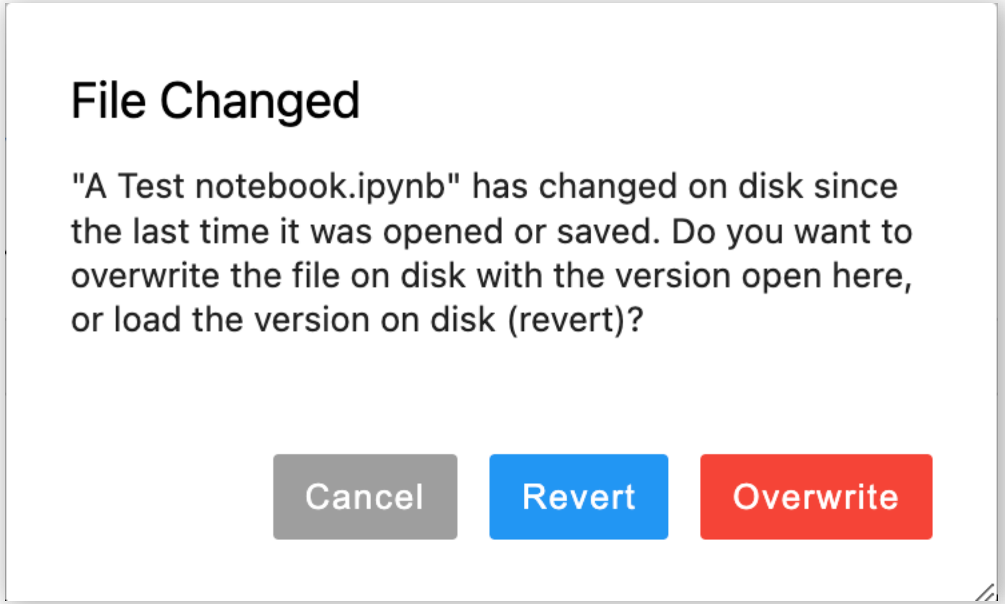
- Similar to a PoC/GSoC project presented in previous years at CS3 Conference

> The integration of BinderHub is on hold for now

A note on collaboration

Current collaboration model for Jupyterlab

- In the beginning, notebooks could not be open in parallel
  - Conflicts would happen, especially on shared filesystems
- Now they can, and their data structures are synchronized
  - This looks awesome!
  - But optimal usage requires sharing the same Jupyter server and kernel (?)
- Jupyterhub proposes “collaboration accounts” instead
  - “Real-time collaboration without impersonation"



File Changed

"A Test notebook.ipynb" has changed on disk since the last time it was opened or saved. Do you want to overwrite the file on disk with the version open here, or load the version on disk (revert)?

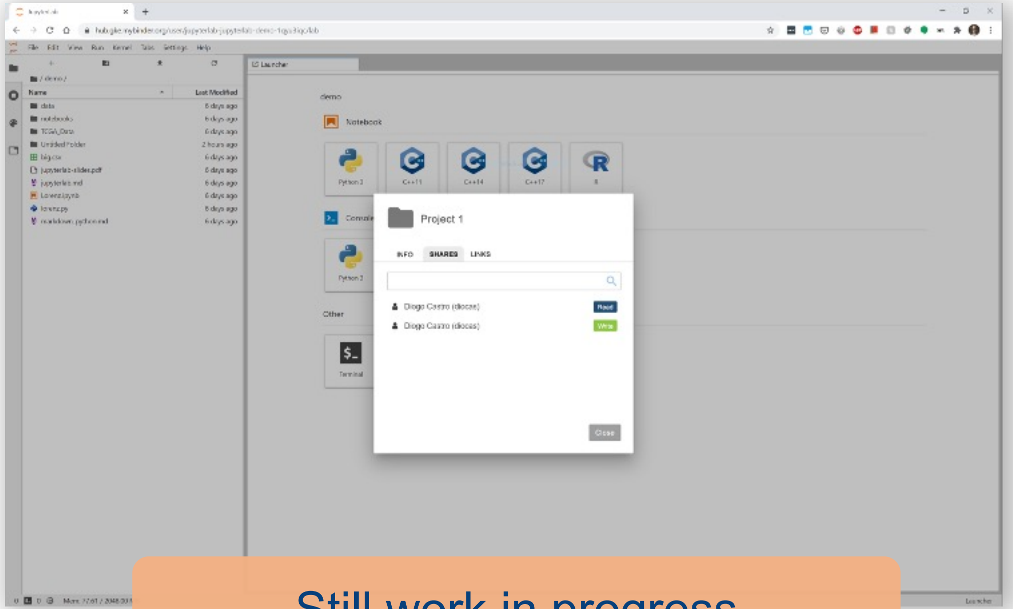
The problems of the current collaboration model

- A shared filesystem might mean access from different Jupyter servers
  - Or even other applications altogether
  - The concurrent editing does not work fully
- Collaboration requires coordination
  - This might not always be easy, especially if we don't know who is editing on the other side…
- Sharing the same server + kernel is risky
  - Full access to another user's account, storage, and permissions on many resources
  - Collaboration accounts help, but might be harder to coordinate or integrate with deployment
- We're not aware of use cases that would benefit from true concurrent editing

We proposed a complementary model better suited for large scale distributed environments

Collaboration model of the CS3Mesh project

- Same view as EFSS inside Jupyter
  - Access files, different mounts, shares, versions, etc.
- Sharing functionality
  - Share with users or public links
  - Same permissions everywhere
- Parallel access to notebooks
  - As alternative to concurrent editing
  - Opening the same notebook without creating conflicts (both locally or remote)
  - Execution environment independence



hub.gke.mybinder.org/user/jupyterlab-jupyterlab-demo-1qpu3iqo/lab

/ demo /

Name	Last Modified
data	6 days ago
notebooks	6 days ago
TCGA_Data	6 days ago
Untitled Folder	2 hours ago
big.csv	6 days ago
jupyterlab-slides.pdf	6 days ago
jupyterlab.md	6 days ago
Lorenz.ipynb	6 days ago
lorenz.py	6 days ago
markdown_python.md	6 days ago

Project 1

Diogo Castro (dlocas) Read

Diogo Castro (dlocas) Write

Still work in progress

https://github.com/sciencemesh/cs3api4lab

Conclusion

Takeaways

- With more configurable options in the upstream Jupyter project, SWAN is being simplified
  - It results in a project that is better to manage and operate at CERN
  - But also easier to deploy outside of CERN
  - The new docker images and full Kubernetes deployment are examples of that
- SWAN continues to work on its integration with heterogeneous and external resources
  - From GPUs to Htcondor via Dask or HPC
  - A Pilot Analysis Facility is ongoing to validate the demand and applicability to CERN use cases
- The collaboration model of Jupyter would benefit from our input
  - As deployers and developers of large sync&share services/products, we have relevant know-how
  - But we need to organise bahind a single voice

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

Evolving SWAN through simplification

Thank you

Diogo Castro
diogo.castro@cern.ch