Software @ SWAN

D. Castro, E. Tejedor, D. Piparo, P. Mato
E. Bocchi, J. Moscicki, M. Lamanna, P. Kothuri

https://swan.web.cern.ch

May 30th, 2018
Librarian and Integrators Workshop

Introduction

What is SWAN

> Service for Web based ANalysis
  - Provides Jupyter Notebooks on demand

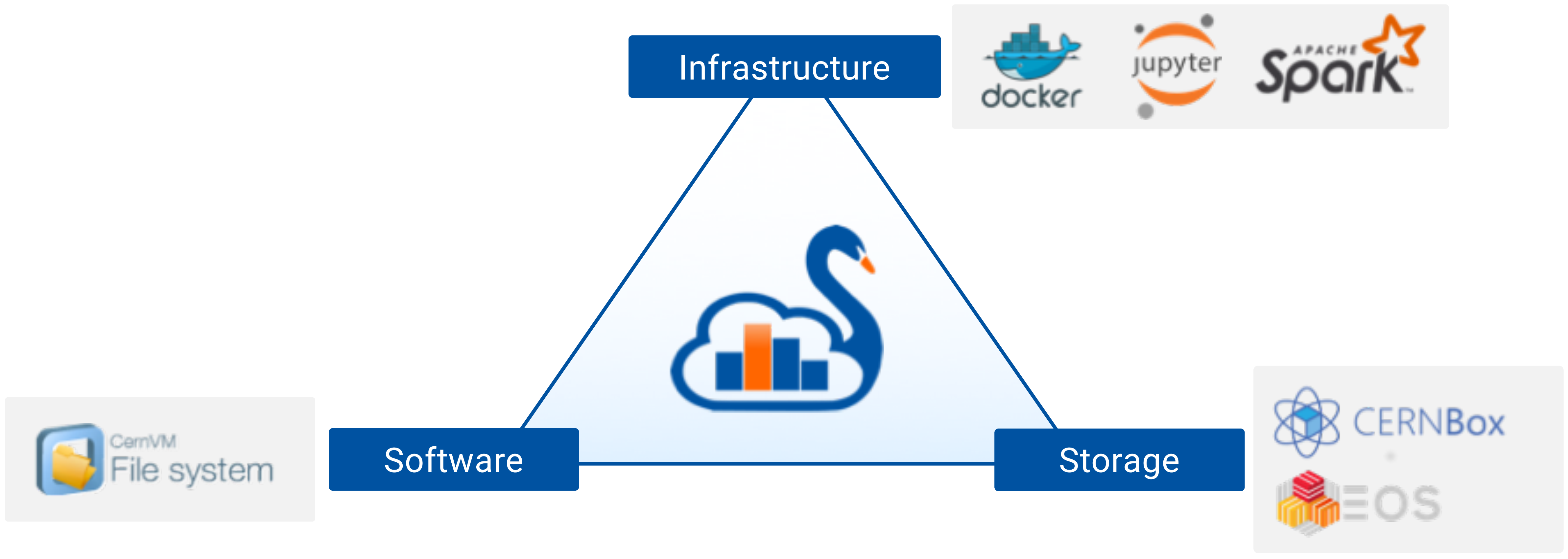
> A web-based interactive interface and platform that combines code, equations, text and visualisations
  - Ideal for sharing/collaboration

> Federates CERN services
  - Including software, storage and infrastructure

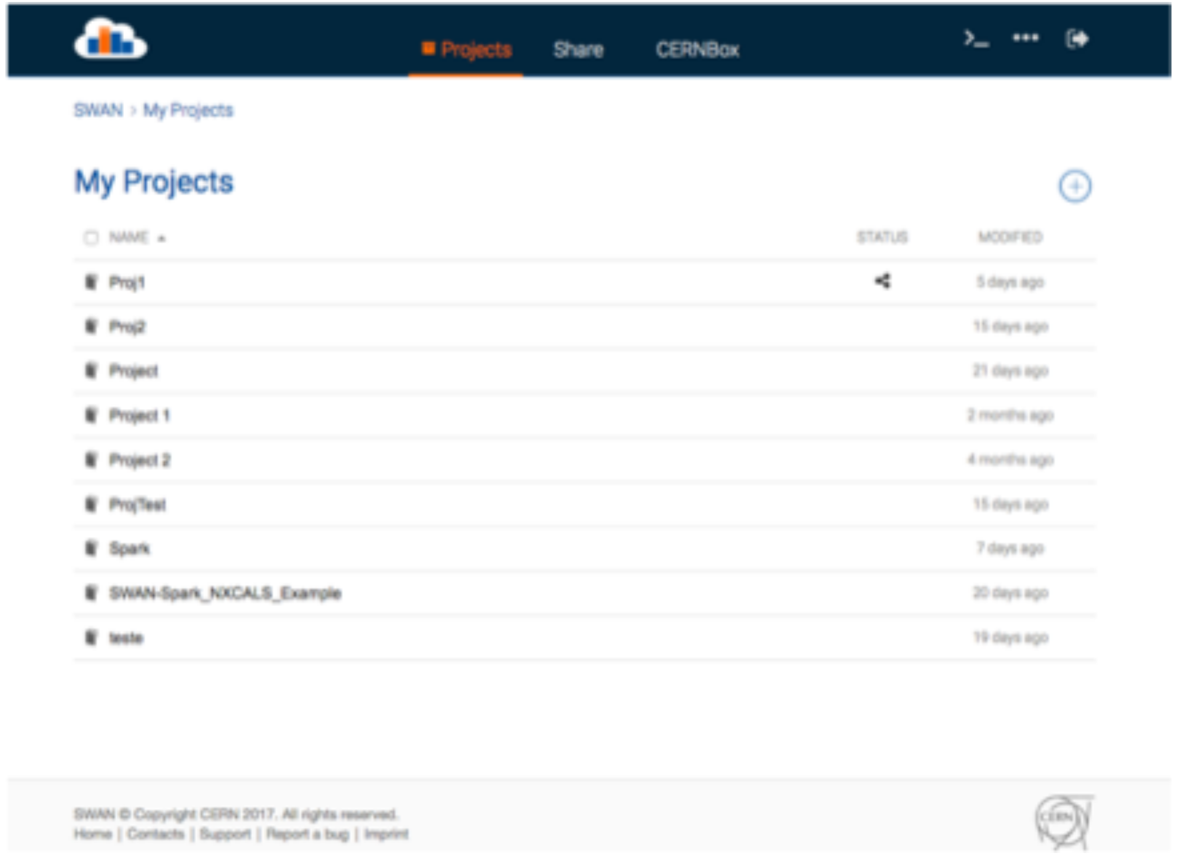
> It's an Interface for Mass Processing Resources
  - For now, Spark integration

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

Integrating services



User Interface

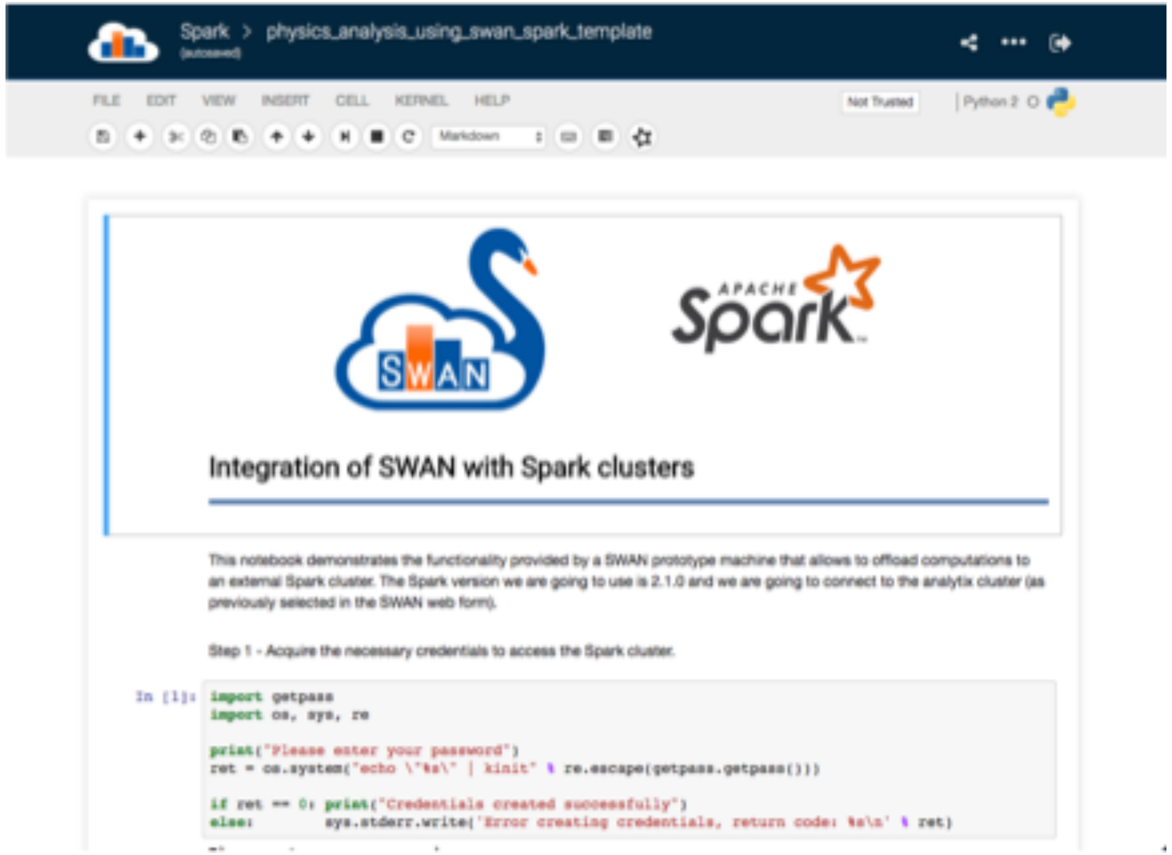


SWAN > My Projects

My Projects

NAME	STATUS	MODIFIED
Proj1		5 days ago
Proj2		15 days ago
Project		21 days ago
Project 1		2 months ago
Project 2		4 months ago
ProjTest		15 days ago
Spark		7 days ago
SWAN-Spark_NXCALS_Example		20 days ago
teste		19 days ago

SWAN © Copyright CERN 2017. All rights reserved.
Home | Contacts | Support | Report a bug | Imprint



Integration of SWAN with Spark clusters

This notebook demonstrates the functionality provided by a SWAN prototype machine that allows to offload computations to an external Spark cluster. The Spark version we are going to use is 2.1.0 and we are going to connect to the analytix cluster (as previously selected in the SWAN web form).

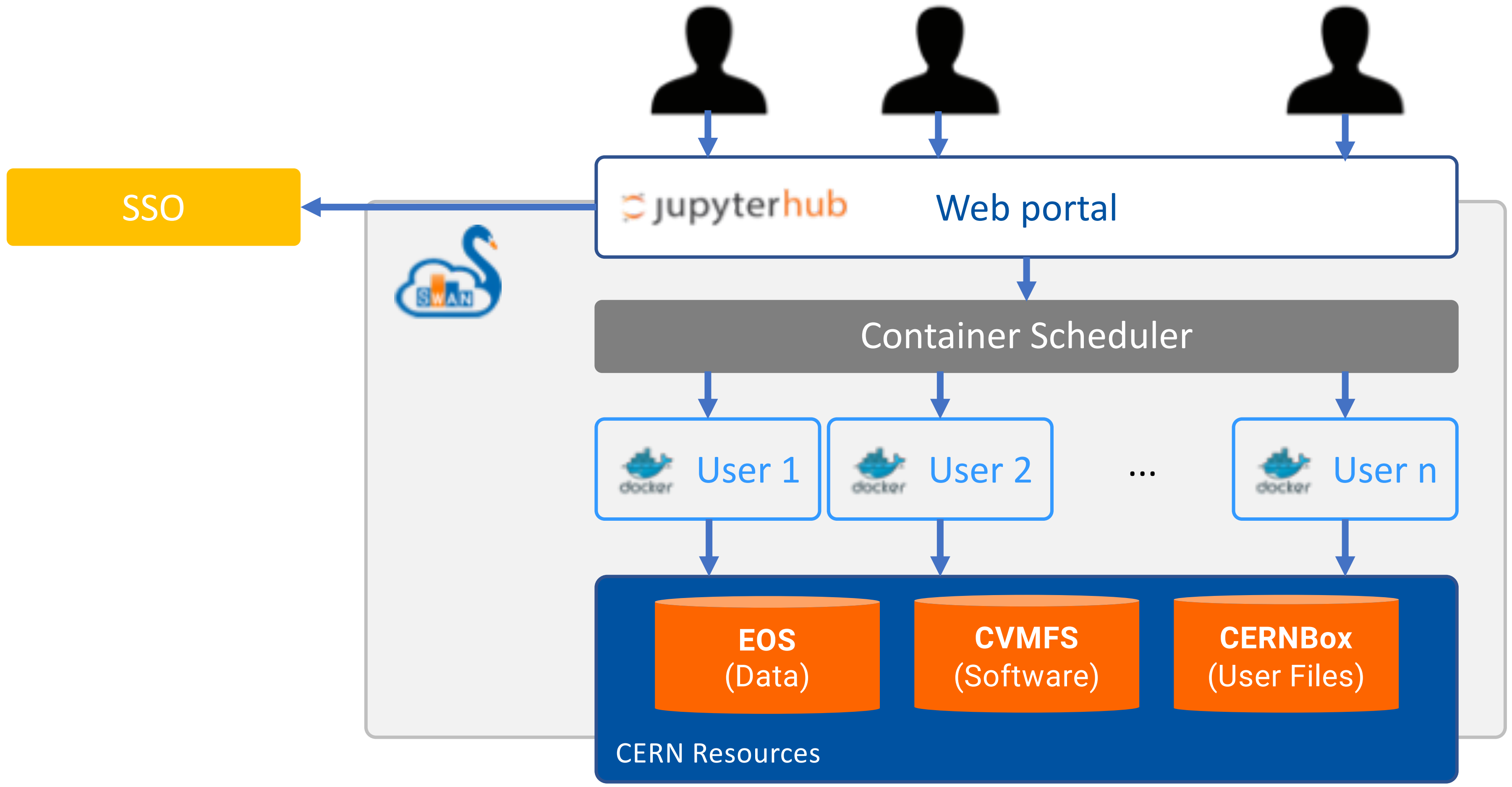
Step 1 - Acquire the necessary credentials to access the Spark cluster.

```
In [1]: import getpass
        import os, sys, re

        print("Please enter your password")
        ret = os.system("echo \"%s\" | kinit" % re.escape(getpass.getpass()))

        if ret == 0: print("Credentials created successfully")
        else:        sys.stderr.write('Error creating credentials, return code: %s\n' % ret)
```

Architecture



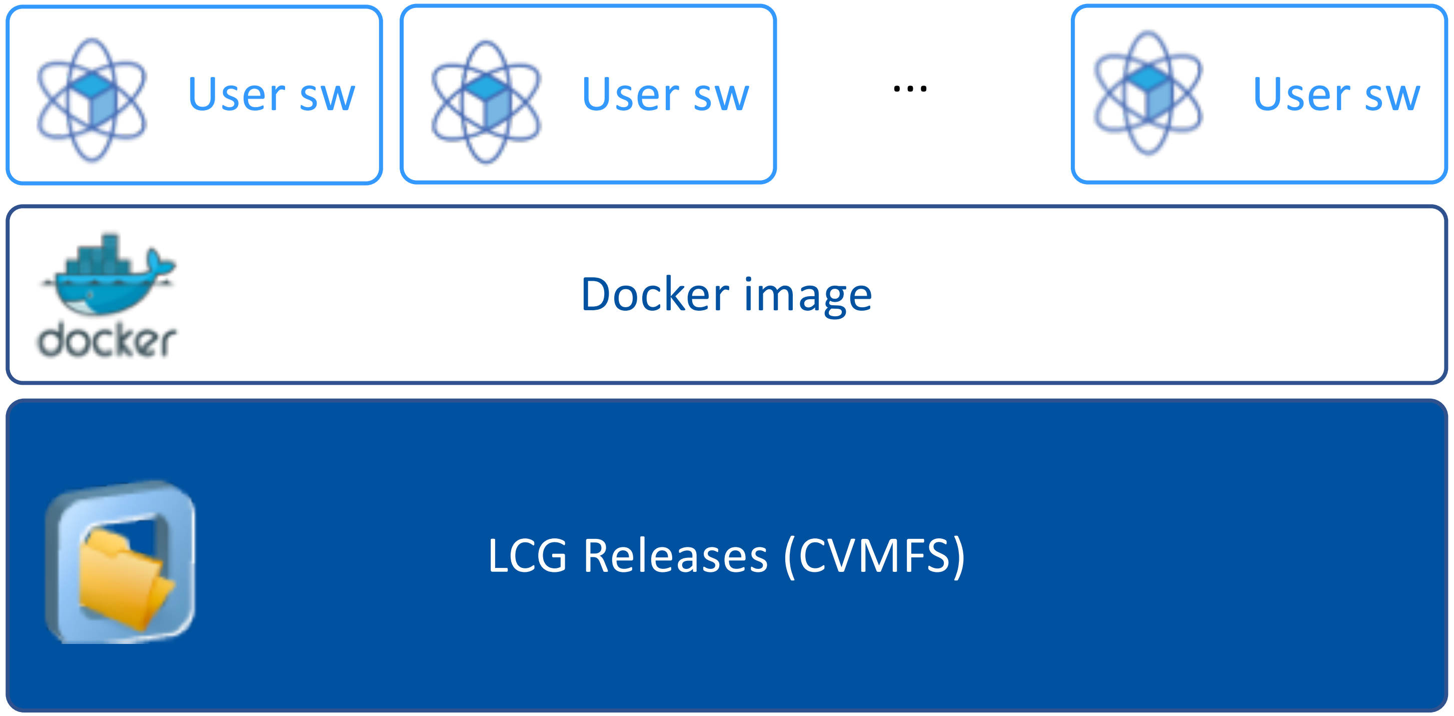
Science Box – Containerized CERN Technology

> Containerized version of all the infrastructure
  - Includes EOS, CERNBox, CVMFS and all Swan services (Jupyter Docker image, JupyterHub)

> Easily deployable on premises
  - Installable in Linux systems
  - Based on Docker Compose
  - "OneClick demo Deployment"
    - https://github.com/cernbox/uboxed
  - "Production oriented Deployment" (orchestration with Kubernetes)
    - https://github.com/cernbox/kuboxed

Software @ SWAN

Where it comes from

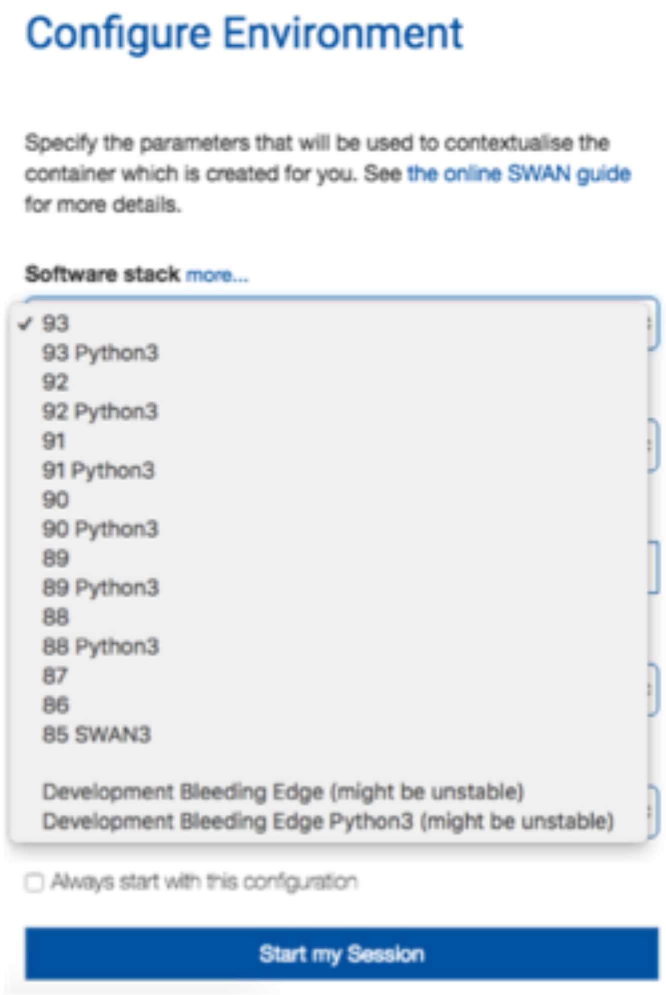


LCG Releases

> SWAN bet on LCG Releases from day 0
  - Removed the need for installation and configuration of packages
  - Reduced the Docker Images size
  - Provides Jupyter kernels

> Multiple stacks
  - Python 2 or Python 3
  - Bleeding edge

> Automatic configuration
  - Set up of notebook and terminal environments



Configure Environment

Specify the parameters that will be used to contextualise the container which is created for you. See the online SWAN guide for more details.

Software stack more...

✓ 93
93 Python3
92
92 Python3
91
91 Python3
90
90 Python3
89
89 Python3
88
88 Python3
87
86
85 SWAN3

Development Bleeding Edge (might be unstable)
Development Bleeding Edge Python3 (might be unstable)

☐ Always start with this configuration

Docker image

> Jupyter modules
  - Ensure compatibility with the whole infrastructure
  - Decouple from user selected stack

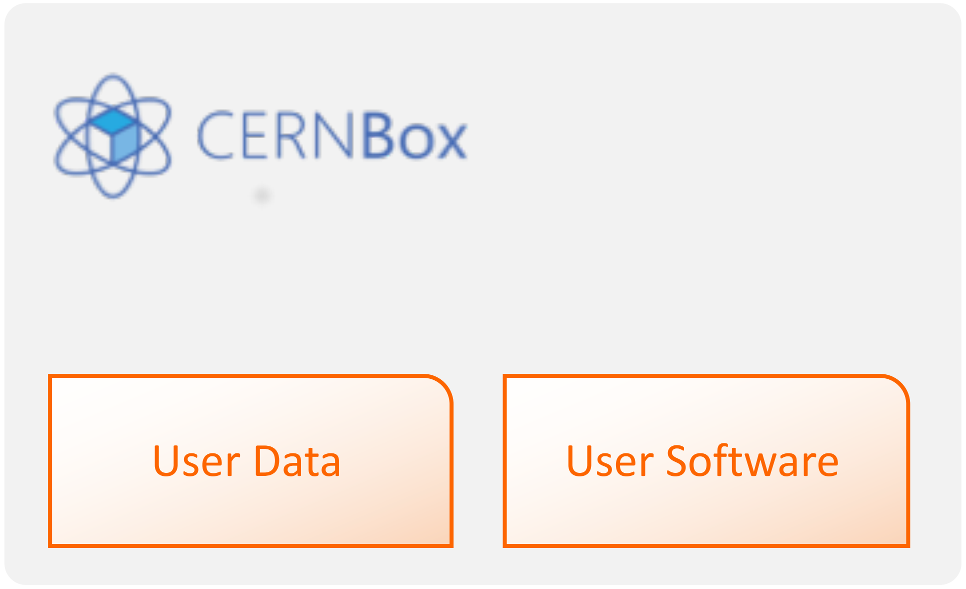
> HEP-Oslibs
  - Same as LXPlus
  - Adds significant weight to the image
  - Could it be distributed by CVMFS?

CERNBox space

> Possibility to install other libraries in CERNBox user storage
  - pip install --user my_package

> Provides a quicker way to use new packages in SWAN
  - Bleeding Edge is daily
  - Releases are ~monthly

> Good way to use custom/not mainstream packages



How new packages are added

1. User sends us an email requesting a new Package

2. We check if it’s valuable for other users

3. We forward the request to the librarians or ask the user to create a Jira ticket

1. Users creates a Jira ticket directly with the librarians
   - This option is advertised in the SWAN Help

In both cases, the packages are added shortly at the Bleeding Edge stack

The Bleeding Edge stack

> Nightlies for CVMFS
  - Good for testing new packages added
  - Good for testing when working on a quick development cycle

> Failures could be detected and reverted
  - Fallback to a previous build

> Stability is key

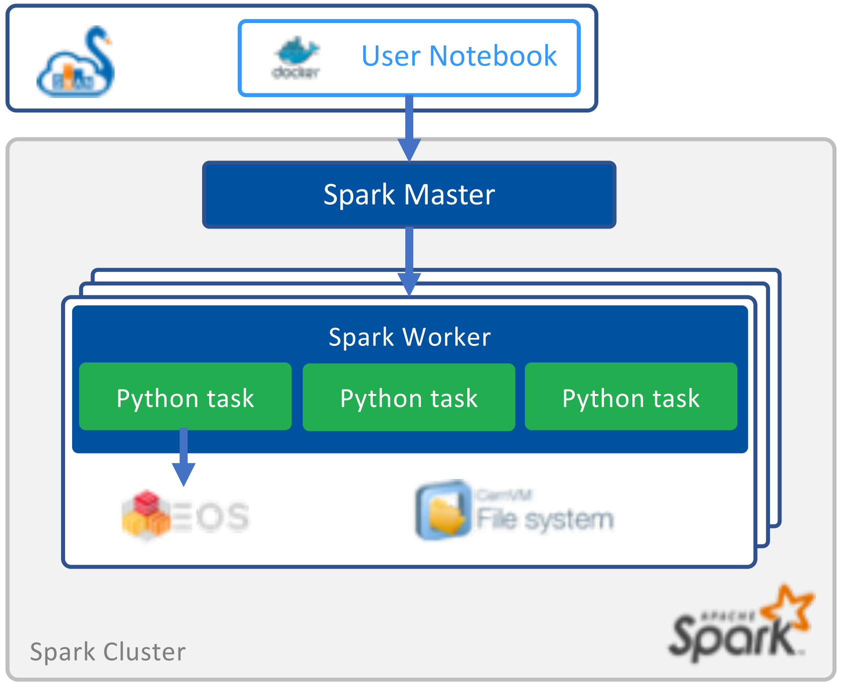
Why LCG Releases are fundamental

Interoperability between services

> SWAN integrates and is an interface for external resources
  - Currently Spark Clusters
  - Working on integration with HTCondor

> Sharing software in a client/server model is a requirement
  - Having the same software stack guarantees that working locally and distributed has the same outcome

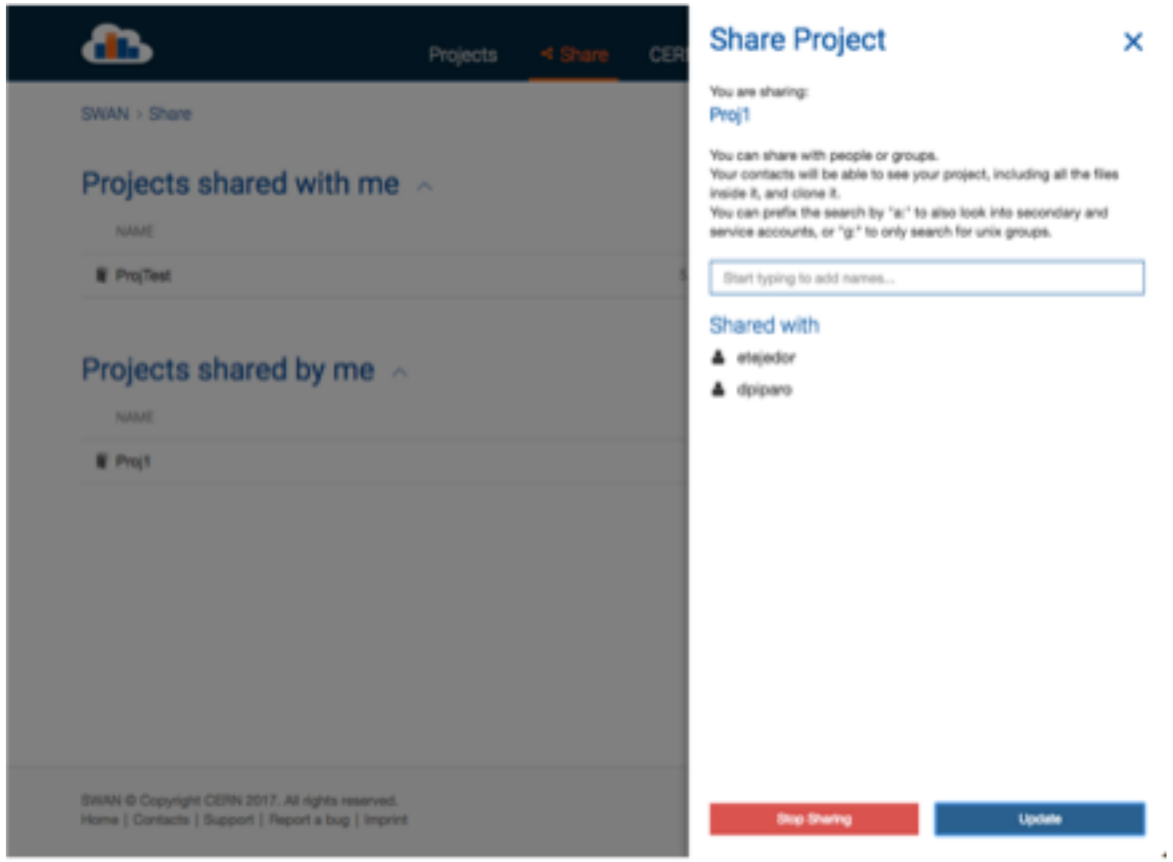
> It facilitates the integration of new services
  - There's only the need to integrate interfaces



Reproducibility

> Sharing of Projects from SWAN is one of the main features of the service
  - Having the same Software stack allows different users to achieve the same results

> LCG Releases are a key ingredient for reproducibility and preservation



You are sharing:
Proj1

You can share with people or groups.
Your contacts will be able to see your project, including all the files inside it, and clone it.
You can prefix the search by "a:" to also look into secondary and service accounts, or "g:" to only search for unix groups.

SWAN © Copyright CERN 2017. All rights reserved.
Home | Contacts | Support | Report a bug | Imprint

Conclusion

Conclusion

> LCG Releases are one of the pillars of the SWAN service

> SWAN allows multiple software sources

- LCG releases: immutable, stable
- Bleeding edge stack: for faster development, availability is still important
- CERNBox user space: immediateness, custom packages

> LCG Releases are a crucial asset to provide interoperable services

Software @ SWAN

Thank you