

Evolution of the SWAN service



Diogo Castro

On behalf of the SWAN team

<https://cern.ch/swan>

Oct 11th, 2019

SWAN Users' Workshop



This workshop is for the SWAN team to learn more about your use cases (...) hearing about how you use SWAN will help us understand how the service should evolve to suit your needs.

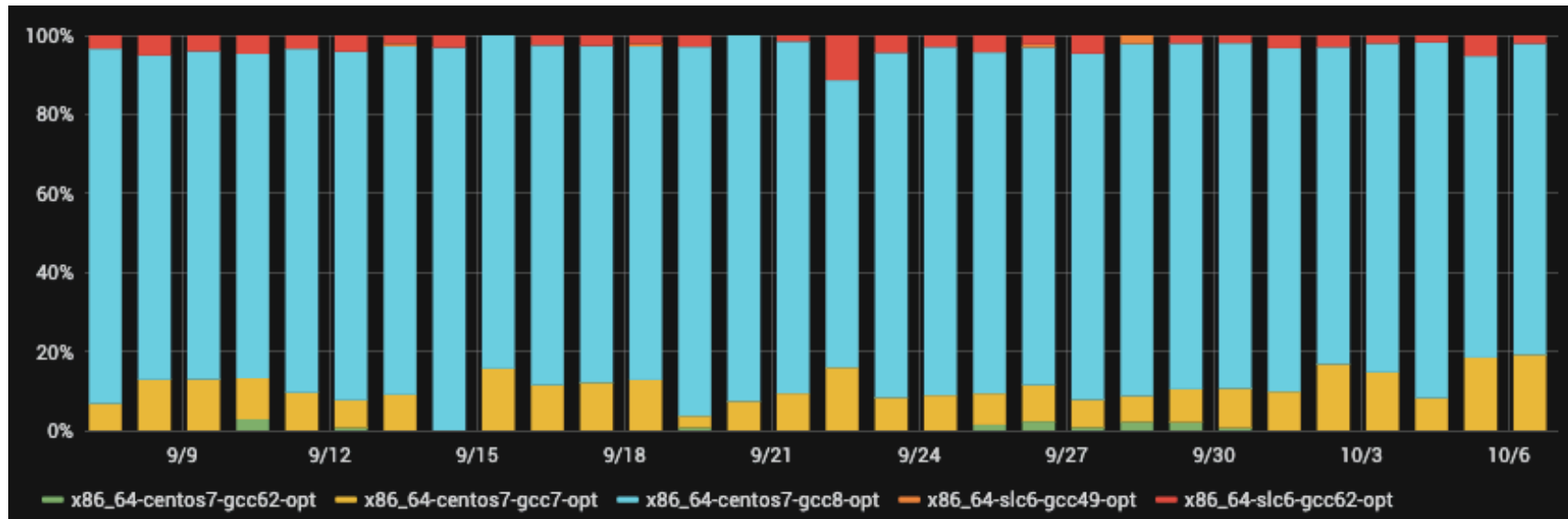
Removals



End of life for SLC6

> January 2020

- Aligned with Ixplus





Deprecation of Python 2

- > January 2020
- > Python 3 as default
 - Old python 2 stacks still available



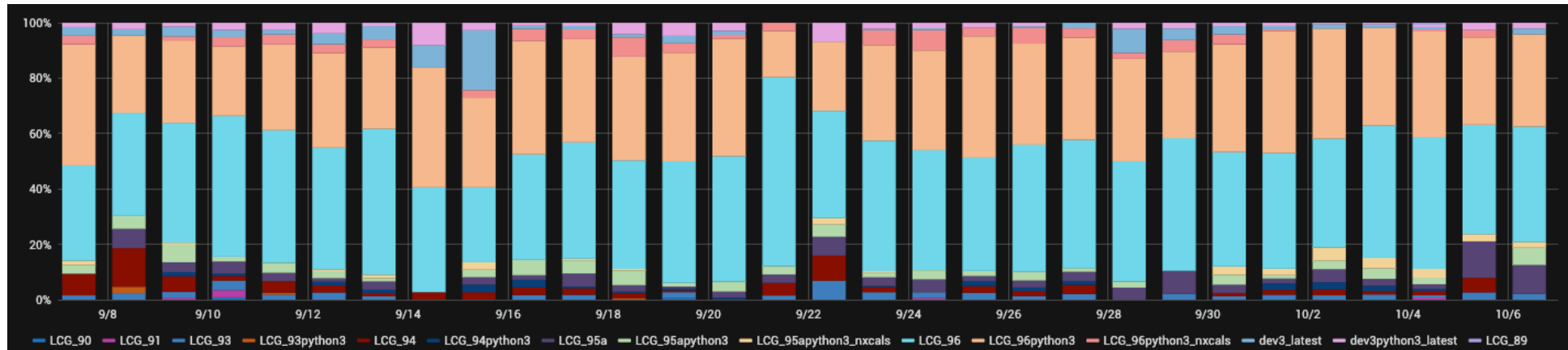
Deprecation of old LCG stacks

> Ongoing discussion

- Depending on support from users

> Stacks with low/no usage in the last 30 days:

- LCG 92 (Python 3), LCG 91 (Python 3), LCG 90 (Python 3), LCG 89 (Python 3), LCG 88 (Python 3), LCG 87, LCG 86, LCG 85



Additions



Jupyterlab

- > Next-generation interface for Project Jupyter
 - Concurrent editing
- > Missing: porting the current extensions
- > Notebooks interface available in parallel during the transition

The screenshot displays the JupyterLab interface. On the left, a sidebar shows a file browser with a list of notebooks and files, including 'Lorenz.ipynb' which is currently selected. The main area is divided into several panes: a top pane with a code editor for 'Lorenz.ipynb' containing text and mathematical equations; a middle pane with an 'Output View' showing a 3D plot of the Lorenz attractor; and a bottom pane with a code editor for 'lorenz.py' containing Python code. The code defines a function 'solve_lorenz' and a derivative function 'lorenz_deriv'. The 3D plot shows a complex, swirling trajectory in a 3D space.

In this Notebook we explore the Lorenz system of differential equations:

$$\begin{aligned}\dot{x} &= \sigma(y - x) \\ \dot{y} &= \rho x - y - xz \\ \dot{z} &= -\beta z + xy\end{aligned}$$

Let's call the function once to view the solutions. For this set of parameters, we see the trajectories swirling around two points, called attractors.

```
In [4]: from lorenz import solve_lorenz
t, x_t = solve_lorenz(N=10)
```

sigma: 10.00
beta: 2.67
rho: 28.00

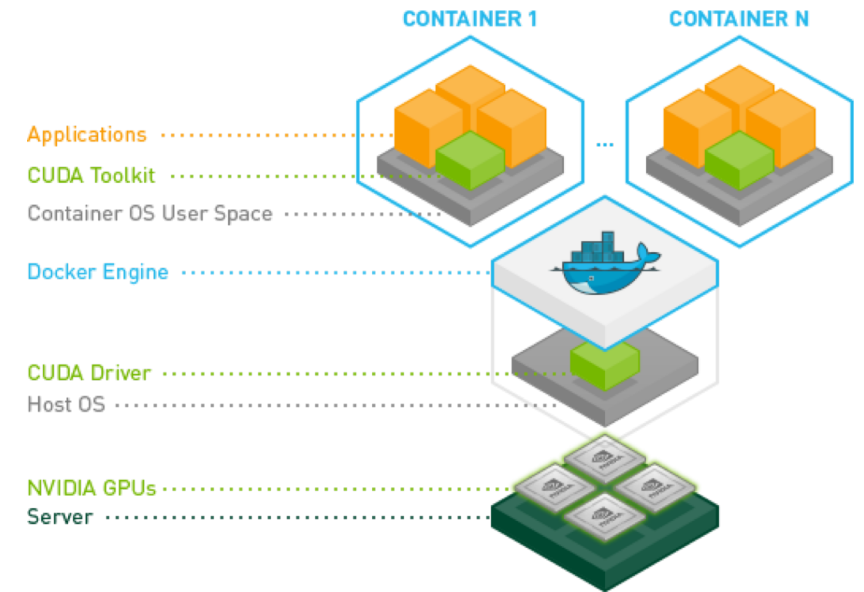
```
9 def solve_lorenz(N=10, max_time=4.0, sigma=10.0, beta=8./3, rho=28.0):
10     """Plot a solution to the Lorenz differential equations."""
11     fig = plt.figure()
12     ax = fig.add_axes([0, 0, 1, 1], projection='3d')
13     ax.axis('off')
14
15     # prepare the axes limits
16     ax.set_xlim((-25, 25))
17     ax.set_ylim((-35, 35))
18     ax.set_zlim((5, 55))
19
20     def lorenz_deriv(x,y,z, t0, sigma=sigma, beta=beta, rho=rho):
21         """Compute the time-derivative of a Lorenz system."""
22         x, y, z = x,y,z
23         return [sigma * (y - x), x * (rho - z) - y, x * y - beta * z]
24
25     # Choose random starting points, uniformly distributed from -15 to 15
26     np.random.seed(1)
27     x0 = -15 + 30 * np.random.random((N, 3))
28
```





NVIDIA GPU Support

- > Exploitation of container technologies to provide support for NVIDIA GPUs
 - Already integrated with ScienceBox
- > Prototype server for testing purposes
 - NVIDIA Tesla V100 PCIe 32GB
 - If interested, ask us to join the beta program
- > All the packages are provided by CVMFS
 - Including CUDA enabled machine learning software stack
 - TensorBoard for interactive monitoring



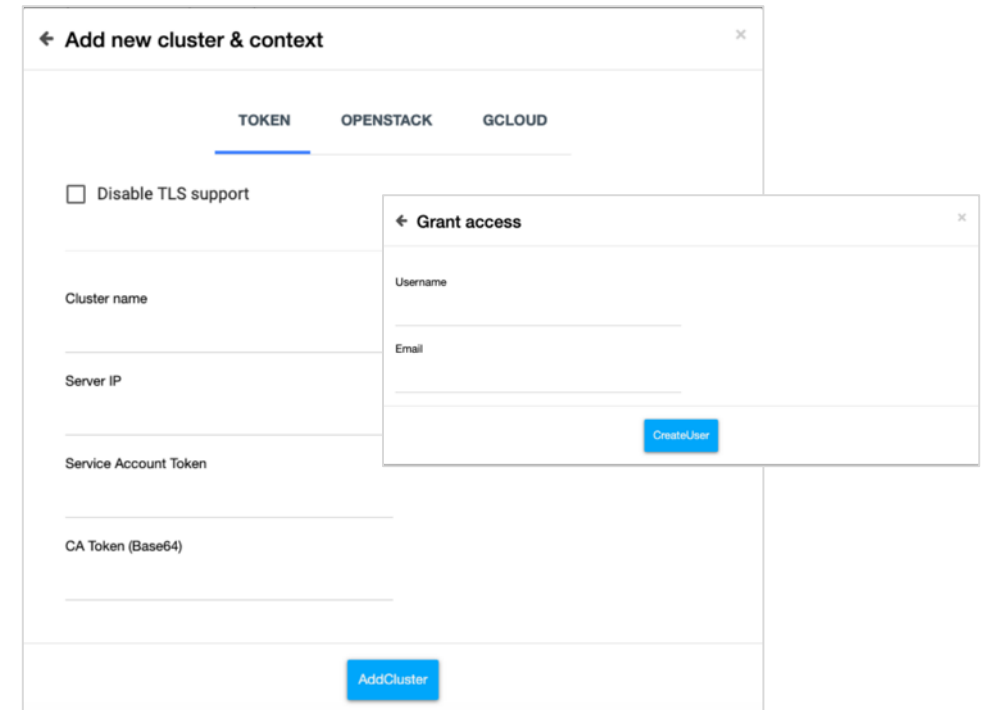
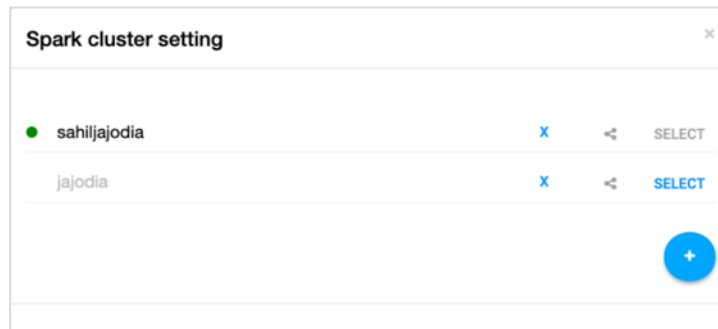


User managed Spark k8s

> Possibility to connect to user managed Kubernetes clusters

- Offload Spark computations
- Control and use your own resources
- Quickly create, use and dispose

> Share access with other users



In exploration



Batch submission

> Ongoing effort: submit **batch jobs** from the notebook

- Using Ganga
- Monitoring display
- Jobs tab



Select jobs to perform action on them. Job ID Jobs per page ↻

<input type="checkbox"/> 0	Job ID	Job Name	Backend	Application	File Name	Status	Submission Time	Runtime
<input type="checkbox"/>	3		Condor	Executable		SUBMITTING	Jul 24th, 3:26 pm	-
<input type="checkbox"/>	2		Condor	Executable		NEW	-	-
<input type="checkbox"/>	1		Localhost	Executable		NEW	-	-
<input type="checkbox"/>	0		Localhost	Executable		COMPLETED	Jul 24th, 3:21 pm	00 seconds

```
In [6]: %%ganga
j=ganga.Job()
j.submit()
```

Backend: LOCALHOST Application: EXECUTABLE Splitter: None 0 SUBJOBS

Job ID	Job Name	Status	Subjobs	Submission Time	Runtime
0		COMPLETED	No Subjobs	Jul 24th, 3:21 pm	00 seconds





Configurable software environment

- > Adding support for Conda environments
 - Linked to Projects
 - Sharable
- > Easy installation of extra packages
 - Clone/import Projects and install the software automatically
- > Still a proof of concept
 - Integration with EOS is starting



The screenshot shows the SWAN web interface. The main area displays a project named 'test1' with a table of files:

NAME	SIZE
Untitled.ipynb	1.03 kB

Overlaid on the right is a 'Configure Project' dialog box. It has a search bar for available packages and two sections:

- Packages To Install:** A table with one entry: 'numpy' with version '1.17.0'. A green 'Install Packages' button is below.
- Installed Packages:** A list of installed packages with their versions, including: '_libgcc_mutex' (0.1), 'backcall' (0.1.0), 'bzip2' (1.0.8), 'ca-certificates' (2019.6.16), 'certifi' (2019.6.16), 'decorator' (4.4.0), 'ipykernel' (5.1.2), 'ipython' (7.7.0), 'ipython_genutils' (0.2.0), 'jedi' (0.15.1), 'jupyter_client' (5.3.1), 'jupyter_core' (4.4.0), 'libffi' (3.2.1), 'libgcc-ng' (9.1.0), 'libsodium' (1.0.17), and 'libtinfo' (6.1.0).

At the bottom of the interface, there is a footer: 'SWAN © Copyright CERN 2016-2019. All rights reserved. Home | Community | Support | Report a bug'.



Changes



Move to Kubernetes

> Ongoing effort

- Based on ScienceBox and upstream
- Pilot infrastructure already in testing

> Improve and modernize SWAN infrastructure

- Replicated, highly-available containers
- Add capacity in minutes to support spikes in service utilization
- Ability to roll out updates with no impact on service capacity
- Leverage on Cloud Containers service by IT-CM



Future improvements

Thank you

Diogo Castro
diogo.castro@cern.ch