

JupyterHub as a service using Docker and Singularity

Azat Khuziyakhmetov

GWGD

Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen

13.12.2021

NHR Container Workshop 2021

JupyterHub

The main goal is to have an easy interface for Jupyter users in HPC.

- JupyterHub¹ implements spawning notebooks on demand for users
- Jupyter notebooks support various kernels: Python, R, Julia etc.²
- IPython Parallel³ can be used to run parallel workers on whole HPC.
- User defined conda environment could be used as a kernel.

¹JupyterHub. URL: <https://jupyter.org/hub>.

²Jupyter kernels. URL: <https://github.com/jupyter/jupyter/wiki/Jupyter-kernels>.

³IPython Parallel. URL: <https://ipyparallel.readthedocs.io/en/latest/>.

JupyterHub in Containers

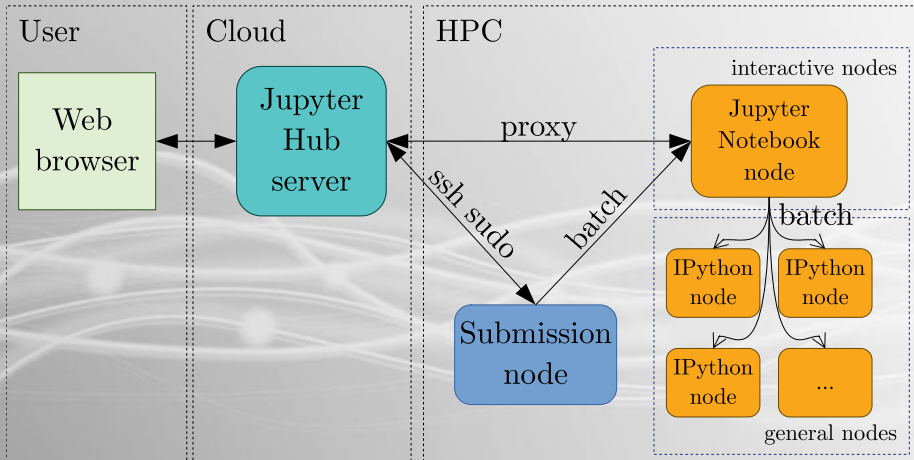
Benefits of running JupyterHub backend with Docker

- There is already Docker containers for JupyterHub deployment
- Can be deployed on any node/server in HPC/Cloud
- Other benefits of running web services in Containers

Benefits of using Jupyter notebook in a Singularity

- There are ready to use containers with preconfigured scientific Jupyter
- Can run multiple versions of Jupyter with the same hub
- The necessary environment for Jupyter is encapsulated
- Containers with a Jupyter notebook can be prepared by users

Implementation in GWDG



Workarounds

The JupyterHub server runs in the cloud without Batch system

- Extended batchspawner⁴ to allow submission from the submission node
- On the submission node there is a script with sudo, which only allows to switch the user and submit/kill predefined notebook job.
- Singularity image definition file⁵ is public to allow users extend it and run their own containers

⁴Batch spawner. URL: <https://github.com/jupyterhub/batchspawner/tree/master>.

⁵Notebook def. URL: https://docs.gwdg.de/doku.php?id=en:services:application_services:jupyter:hpc#running_your_own_singularity_container.

Live Demo

Example - spawn

Spawner Options

Select a job profile:

Set the duration (in hours):

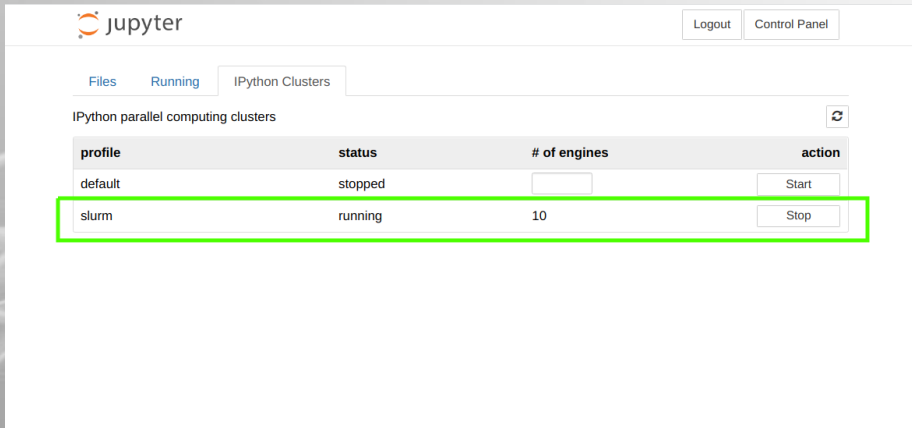
Set the number of cores:

Set the amount of memory (in GB):

Jupyter Notebook's Home directory

[Documentation](#)

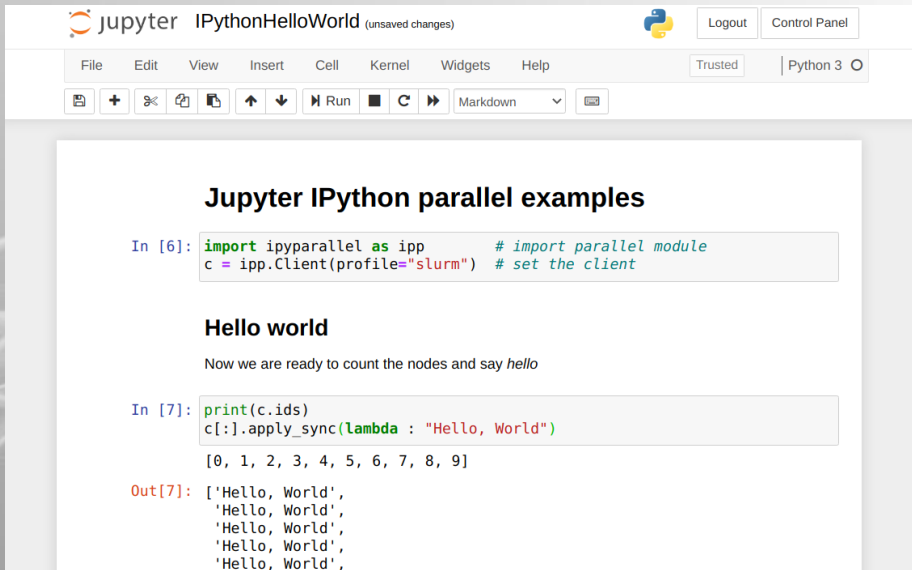
Example - set cluster



The screenshot shows the JupyterLab interface with the 'IPython Clusters' tab selected. The interface includes a 'Logout' button and a 'Control Panel' button in the top right. Below the navigation tabs, there is a section for 'IPython parallel computing clusters' with a refresh icon. A table lists the clusters, with the 'slurm' cluster highlighted by a green border. The table has columns for profile, status, # of engines, and action.

profile	status	# of engines	action
default	stopped	<input type="text"/>	Start
slurm	running	10	Stop

Example - run



The screenshot shows a Jupyter Notebook interface with the following elements:

- Header: "jupyter IPythonHelloWorld (unsaved changes)" with a Python logo, "Logout", and "Control Panel" buttons.
- Menu: "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", "Help".
- Trust: "Trusted" button and "Python 3" dropdown.
- Toolbar: Includes icons for home, add, undo, redo, save, run, clear, and a "Markdown" dropdown.
- Content:
 - Jupyter IPython parallel examples**
 - In [6]:** `import ipyparallel as ipp # import parallel module`
`c = ipp.Client(profile="slurm") # set the client`
 - Hello world**
 - Text: "Now we are ready to count the nodes and say *hello*"
 - In [7]:** `print(c.ids)`
`c[:].apply_sync(lambda : "Hello, World")`
 - Output: `[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]`
 - Out[7]:** `['Hello, World',`
`'Hello, World',`
`'Hello, World',`
`'Hello, World',`
`'Hello, World']`



Conclusion

In general JupyterHub in HPC is a well received service.
Weekly usage in GWWDG local HPC is ~ 20 unique users.

Advantages

- Easier access to HPC for users familiar with Jupyter
- Can be used as a frontend for other services
- Containers help for fast deployment

Disadvantages

- Notebook containers should be updated along with the software on HPC
- If functionality of containers is extended, should be compatible with the upstream version

Thank you!

A decorative graphic at the bottom of the slide consisting of several overlapping, glowing white wavy lines that create a sense of motion and depth against the light grey background.