



JupyterHub: a Web Based Programming Environment for Teaching and Learning

Debsankha Manik, Jana Lasser

Max Planck Institute for Dynamics and Self-Organization (MPIDS),
Am Fassberg 17, 37077 Göttingen, Germany

debsankha.manik@ds.mpg.de, jana.lasser@ds.mpg.de

The Problem

- Need to set up identical programming environment for every user (for e.g. teaching, workshops, data analytics teams).
- Should work on any client machine, independent of OS and hardware.
- Adding packages/libraries should be easy.
- Should plug into existing authentication systems.
- Should be scalable if number of users increase.

The Solution: JupyterHub

- Users need nothing more than a web browser.
- Each user gets identical tools, libraries (specified via a docker image).
- Scales for high number of users using kubernetes.
- Integration into existing authentication system (LDAP, GitHub, GitLab etc).
- Support for more than 20 programming languages.
- Jupyter notebook file format: code, plots, images and markdown/L^AT_EX text in the same file.

Jupyter Notebook: Code, Plots and Paper in a Single File

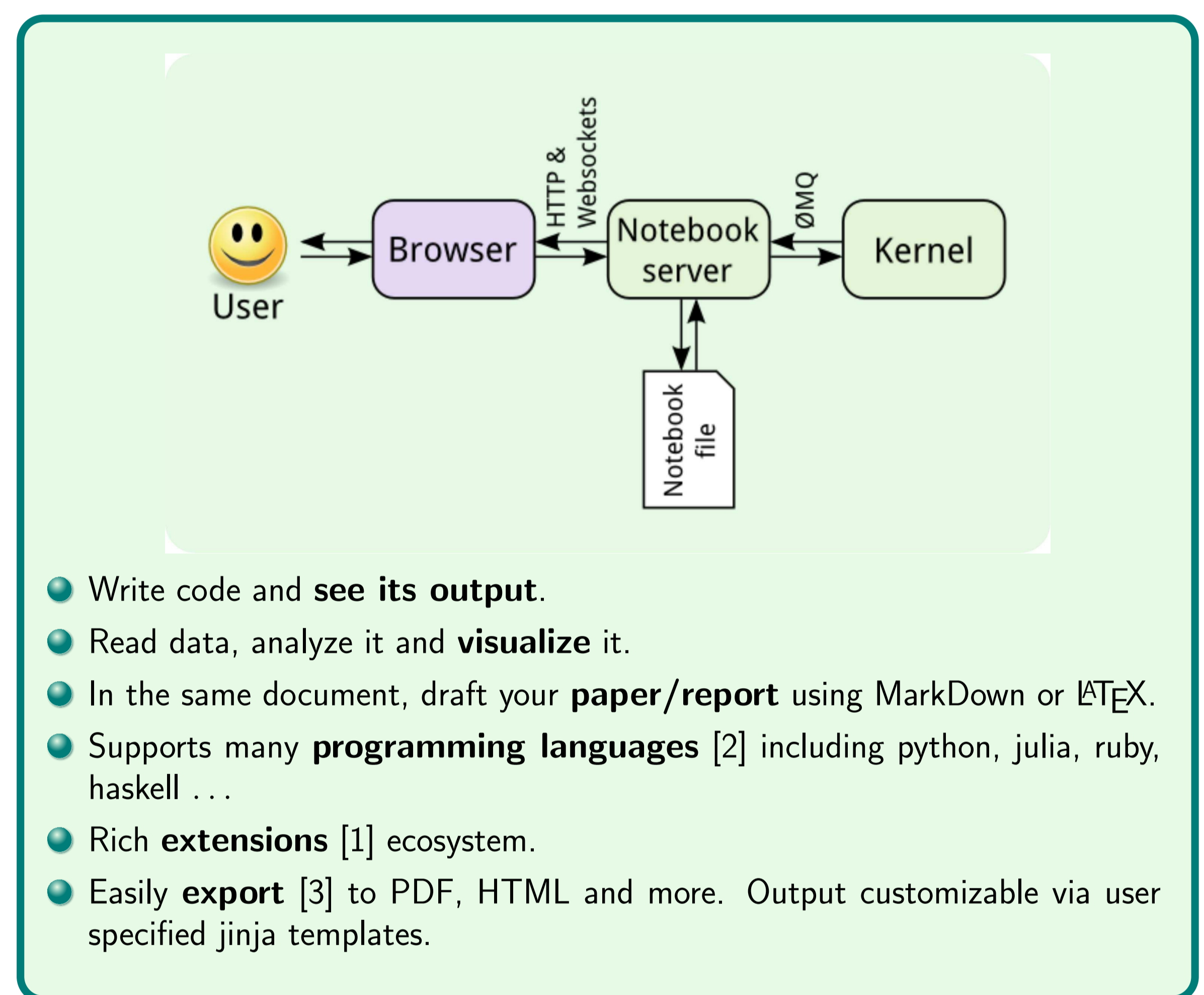
The screenshot shows a Jupyter Notebook interface in a browser. The top bar includes the Jupyter logo, the notebook name 'hello_world', and a 'Last Checkpoint' timestamp. Below the toolbar, there is a text area with instructions on how to use a Jupyter Notebook. The main area contains a code cell with the following code:

```
In [3]: 1 + 1
Out[3]: 2

In [4]: import seaborn as sns
import pandas as pd
import matplotlib inline

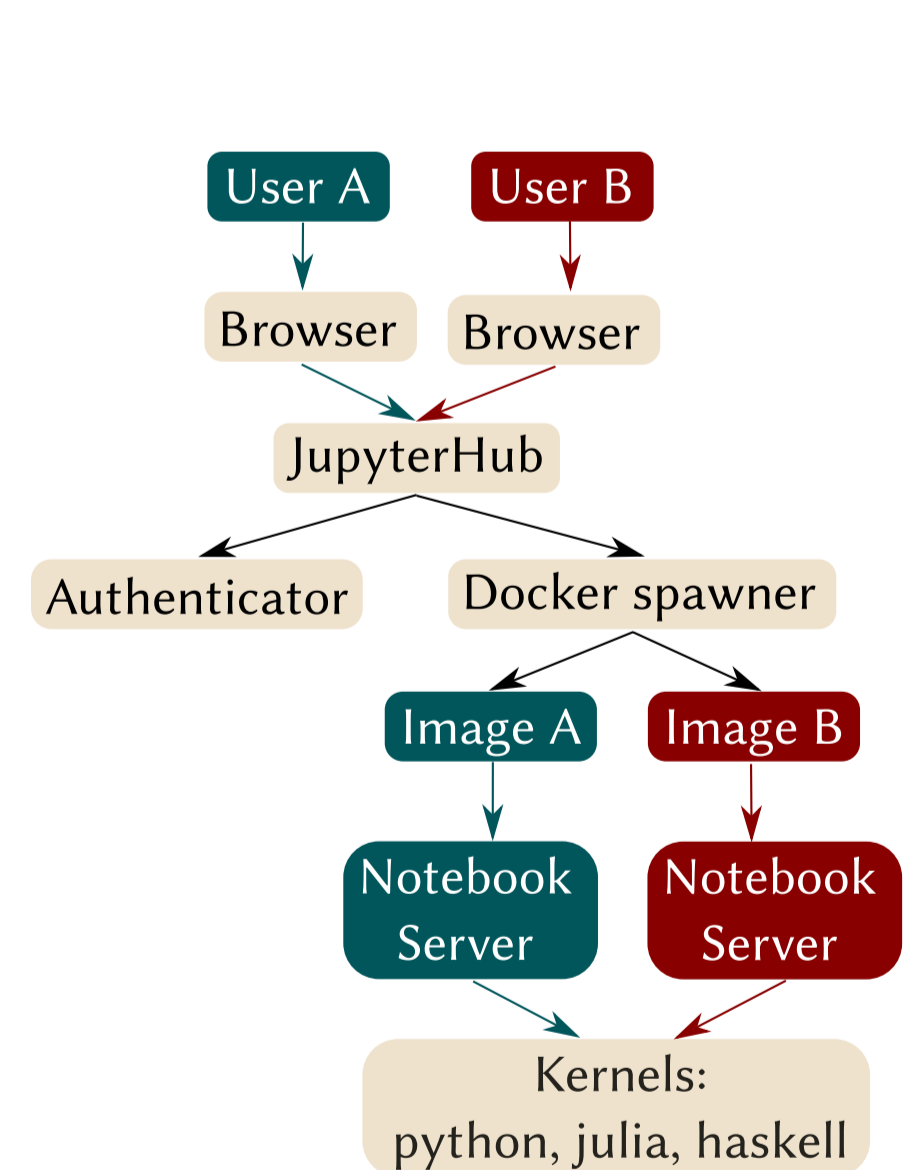
In [5]: iris = sns.load_dataset('iris')
sns.pairplot(iris)
Out[5]: <seaborn.axisgrid.PairGrid at 0x7fb55ddb3198>
```

Below the code, there is a pair plot of the iris dataset, showing a histogram of sepal length and scatter plots of sepal length vs. other variables.



- Write code and **see its output**.
- Read data, analyze it and **visualize** it.
- In the same document, draft your **paper/report** using Markdown or L^AT_EX.
- Supports many **programming languages** [2] including python, julia, ruby, haskell ...
- Rich **extensions** [1] ecosystem.
- Easily **export** [3] to PDF, HTML and more. Output customizable via user specified jinja templates.

JupyterHub: Jupyter Notebooks as a Service



- Each user gets own **docker container** and notebook server.
- Docker image exactly specifies the tools users have access to.
- **Docker volumes**: each user can access a shared filesystem.
- Multiple **authenticators** [4] are supported: Bitbucket, GitHub, GitLab, Google, MediaWiki, OpenShift etc.
 - Easy to write ones own authenticator.
- **Resource allocation**: each user can be allocated X GB memory.
- **Multiple kernels** can be simultaneously available: python, haskell, julia ...

Our Setup

- A data literacy course open to **all departments** in University of Göttingen [8].
- **60 participants** on first iteration.
- JupyterHub provides the computing environment.
- Deployed by **GWDG** on a VMWare Cluster.
- Using the docker image `jupyter/datascience-notebook`, provides many popular tools in python data science ecosystem (numpy, matplotlib, pandas, seaborn ...)
- Each user allocated 4 GB RAM.
- CPU and RAM hotplug supported for dynamic scaling up.

Outlook

JupyterHub at scale

- Scale to large number of users: deploy on a kubernetes [5] cluster.
- Scale to a HPC cluster
 - Integrate with a queuing system [9] like Sun Grid Engine.
 - ipyparallel [6] for easy parallelization.

Feature wishlist

- Collaboratively working on Jupyter notebooks.
- Easier way to version control (maybe jupyter [7]).

References

- [1] jupyter-contrib-nbextensions.readthedocs.io/en/latest
- [2] github.com/jupyter/jupyter/wiki/Jupyter-kernels
- [3] github.com/jupyter/nbconvert
- [4] jupyterhub.readthedocs.io/en/stable/reference/authenticators.html
- [5] z2jh.jupyter.org/en/latest
- [6] github.com/ipython/ipyparallel
- [7] github.com/mwouts/jupyterx
- [8] uni-goettingen.de/de/592287.html
- [9] info.gwdg.de/docs/doku.php?id=en:services:application_services:jupyter:hpc