Jupyter Notebooks

How to edit and share easily your notebooks

09/11/2023

Pierre-Antoine Bouttier

Who am I?

- Research engineer, software engineering expert, specialized in applied mathematics, ...
- ...Deputy director of the UAR GRICAD, based in Grenoble, providing **services, infrastructure, and expertise to support all Grenoble research communities** in the fields of **scientific computing, software development, and research data management.**
- I am not...
  - ...an SysAdmin
  - ...a computer scientist

TOC

TOC

Jupyter notebooks at a glance

Jupyter notebooks can be seen as :

- electronic notebooks...
- ... accessible via a **browser** ...
- ... made up of different **cells** .

Jupyter notebooks at a glance

Cells can contain :

- formatted text (markdown)
- images
- mathematical formulas
- executable software code

Demo

Jupyter notebooks at a glance

Initially developed for Julia, Python and R (Jupyter), today over 40 languages are supported.

- A notebook can be exported in a variety of formats (HTML, PDF, LaTeX, etc.).
- It's easy to extract only code cells from a notebook.
- You can install `jupyter-notebook` on your machine, and notebooks will run on your machine (through browser access).

How to create and/or edit a notebook?

If you are not familiar with editing and work with notebooks, I propose to:

- Install VSCode, a multi-languages code editor with nice IDE features (Open source but distributed binaries include MS telemetry)...
- ...Or VSCodium, really FLOSS

- Then, add Jupyter and python extensions
- Start to create and edit a notebook

Hands-on

What's in it for me?

IMHO, notebooks are truly relevant tools for **transmitting** knowledge:

- Documenting the use of software code
- Presenting a reasoning or methodology that includes software code
- write enriched articles, tutorials, HOWTOs
- etc.

What shouldn't it be used for?

- Developing software code: code in notebooks is difficult to maintain, be modular, readable, etc. **Supplying a notebook does not, in itself, guarantee reproducibility of the work thus presented! (no intrinsic control on the environment)**
- Providing a notebook does not in itself guarantee the reproducibility of the work thus presented!
- Depending on the execution environment, mediocre performance.

TOC

How do I run a notebook?

On your computer, the classic method:

- Install `jupyterlab` on your computer (or VSCode/VSCodium, for example)
- Install the software libraries required by our notebook code.
- Launch a jupyter server on your computer
- Access the notebooks via the browser (no internet needed, local network)

Some drawbacks

- The client/server interface is confusing for beginners.
- Lots of things to install…
- …which can involve different tools, more or less available depending on the computer.

What about training?

If you want to offer training materials to several students :

- They'll have to do everything on their own:
  - With potentially heterogeneous OSs
  - With heterogeneous skills
- Or you're going to have to set up a notebook server that students can access themselves: tedious and possibly complicated.

TOC

GitLab to the rescue

A good way to share your notebooks is to set up a GitLab project to host them.

For this session, two options:

- Create your own project on GitLab
- Create a GitLab project and Clone this one.

Hands-on

GitLab to the rescue

GitLab makes it easy **to provide notebooks** AND **their runtime environment!**

GitLab to the rescue

GitLab makes it easy **to provide notebooks** AND **their runtime environment!**

In your GitLab project, you need to specify the needed software environment (e.g. `requirements.txt` for Python codes)

GitLab to the rescue

GitLab makes it easy **to provide notebooks** AND **their runtime environment!**

Let's take a look at two options:

- BinderHub
- JupyterLite

BinderHub

BinderHub is a web service that allows on-the-fly deployment of notebook runtime environments from a public GitLab project.

A site based on this service accessible to the general public

Hands-on

Jupyterlite

JupyterLite provides a JupyterLab runtime environment without the client/server architecture (and therefore without the complications that go with it).

- With a single command, jupyterlite generates a static site (simple html pages), along with your notebooks and other files if you wish.
- Using a web server, **this site will run as a jupyterlab interface in the student's browser**.

How is this simpler?

But how do I install and configure a web server?

- GitLab can do it for you!
- Using CI/CD, in the `.gitlab-ci.yml`, simply generate the site with JupyterLite and publish it with GitLab Pages.

GitLab CI/CD, very (very) briefly

Continuous Integration/Continuous Delivery and/or deployment is a mechanism, on a software forge, that runs automatic tasks given triggers.

- `git push` from your local machine to GitLab "updates" your GitLab code repository with new commits.
- If CI/CD is activated on your GitLab projects, by default, it triggers some tasks that you have define.

GitLab CI/CD from a very quick pratical POV

To activate and run CI/CD tasks on a GitLab project, you need:

- Activate it in your project settings on GitLab
- Define a **runner** (i.e. a program that monitors changes on your repo. and launch CI/CD tasks)
- Define a `.gitlab-ci.yml` at the root of your repo., that will describe what are your CI/CD tasks.

Back to the GitLab project to deploy a jupyterlite instance on your GitLab project

TL;DR (1/3)

Notebooks are great tools:

- To demonstrate a methodology
- As a course aid
- To explain how a software code/data processing chain works

TL;DR (2/3)

Notebooks are not suitable for :

- Reproducibility (they don't guarantee it themselves)
- Software development
- In some cases, performance (computing and IO)

TL;DR (3/3)

You have at least 2 methods at your disposal for simply deploying a runtime environment along with your notebooks, easy to share (a simple url to pass on):

- BinderHub
- JupyterLite

Feel free to use this repository as inspiration for deploying notebook runtime environments

Thank you for your attention!