

TP HPC/Big Data

Utilisation de la plateforme Jupyterhub du CNES

SUPAÉRO
MODULE FITR35

Accès au SI Scientifique du CNES


Avec l'identifiant qui vous a été envoyé par mail, choisissez un mot de passe sur le portail : <https://account-sis.cnes.fr> en choisissant "Mot de passe oublié".

Jupyterhub

Depuis un navigateur, se connecter sur l'adresse :

<https://jupyterhub.cnes.fr>

Entrer l'identifiant et le mot de passe précédemment initialisé.



CNES

CENTRE DE CALCUL

Accès au système d'information scientifique du CNES. Obligation de clore votre session lorsque vous quittez votre poste de travail.

Access to the CNES scientific information system. Obligation to close your session when you leave your workstation.

Bienvenue sur le Jupyterhub du Centre de Calcul du CNES !

You can connect with your CNES Scientific Information System user account. Once logged in, you'll have access to a Jupyter notebook server started on a computing node. You'll thus be able to analyse your data or submit computations interactively. For more information, please see the Computing Center Wiki (in french).

Sign in

Username:

Password:

Sign in

Choisir les ressources nécessaires :

- Default Lab environment
- Partition = visu default
- 1 core pendant 4 heures pour les TPs OpenMP et MPI
- 4 cores pendant 4 heures pour le TP Dask

Server Options

Cluster weather:

07/10/2024 14:29

CPU 2022 : 

- Free ressources over CPU 2022: 2.12%
- Maximal number of free CPU 2022 over one node: 19
- Number of free CPU 2022: 225

GPU : 

- Free ressources over GPU: 12.5%
- Number of free GPU: 1

Select the environment:

You can select a Virtual Research Environment (VRE) for starting your Jupyterlab instance.

It usually provides domain or project software and libraries packaged together and ready to use.

Be aware that you can only run VRE if you belong to the associated user group, you will get an error if not.

If you've never heard of VREs, just use the default environment.

For help please contact [HPC support](#).

For help with VRECESWOT, please use this [link instead](#).

☐ Check to use latest version of VRE (might be unstable)

Default lab environnement - without VRE (all groups) ▼

Partition:

Visu (Default) ▼

Ressources:

1 core
7 GB

2 core
14 GB

4 core
28 GB

Job duration:

1
hour

2
hours

4
hours

6
hours

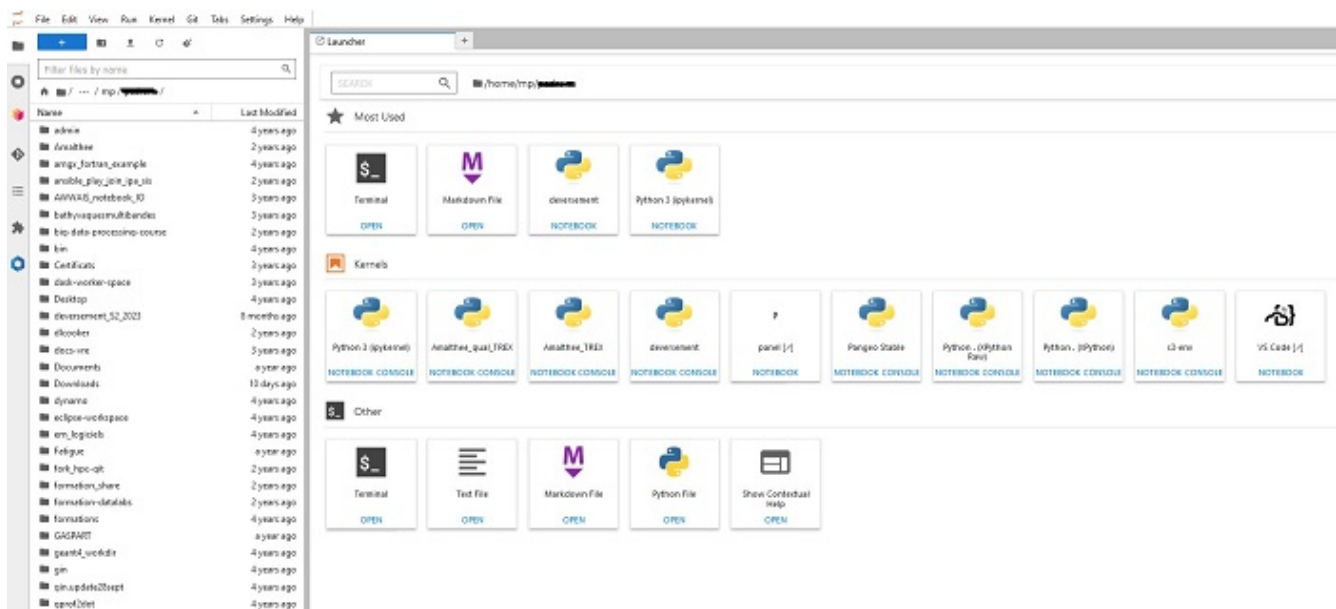
8
hours

10
hours

12
hours

Job will end at 15:37

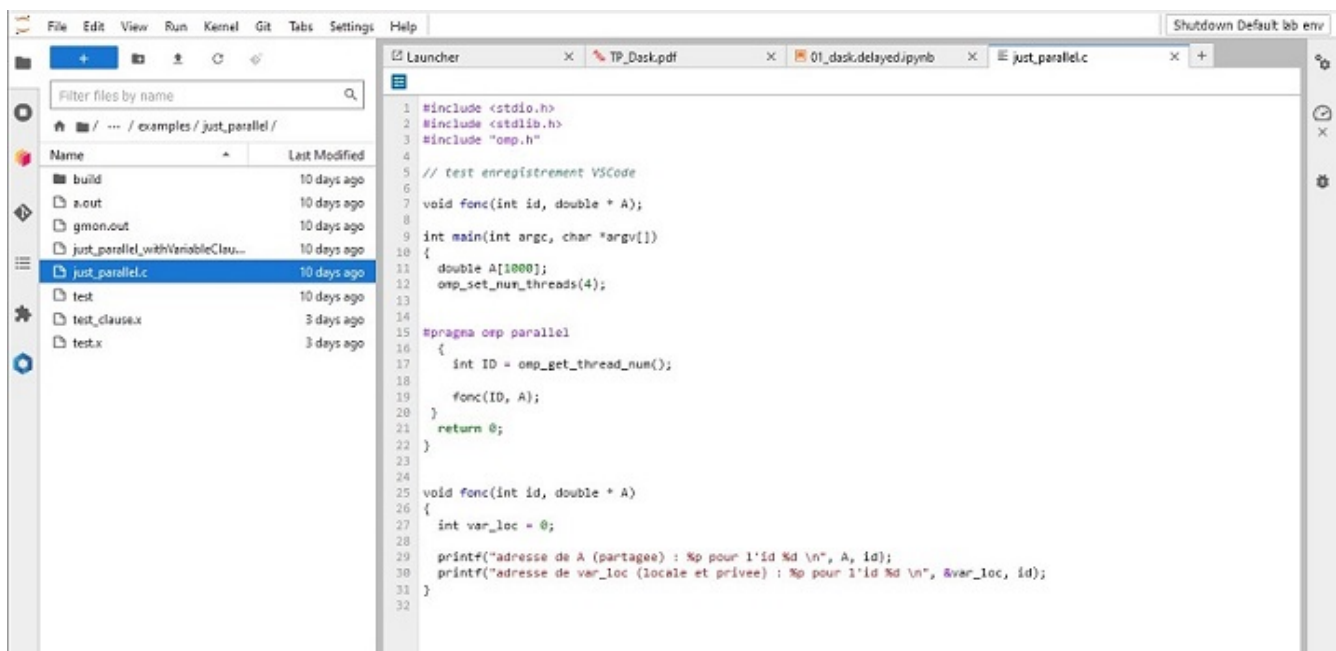
Launcher



Cliquer sur terminal pour avoir un shell sur le noeud de visualisation sur lequel votre session s'est lancée. Ce terminal sera utilisé en permanence pour exécuter les commandes décrites dans les 2 premiers TP : OpenMP et MPI.

1 Utilisation du JupyterLab pour les 1er et 2ème TPs (OpenMP et MPI)

Depuis le bureau Jupyter, il est possible d'afficher le code source en cliquant sur le fichier `.c`.

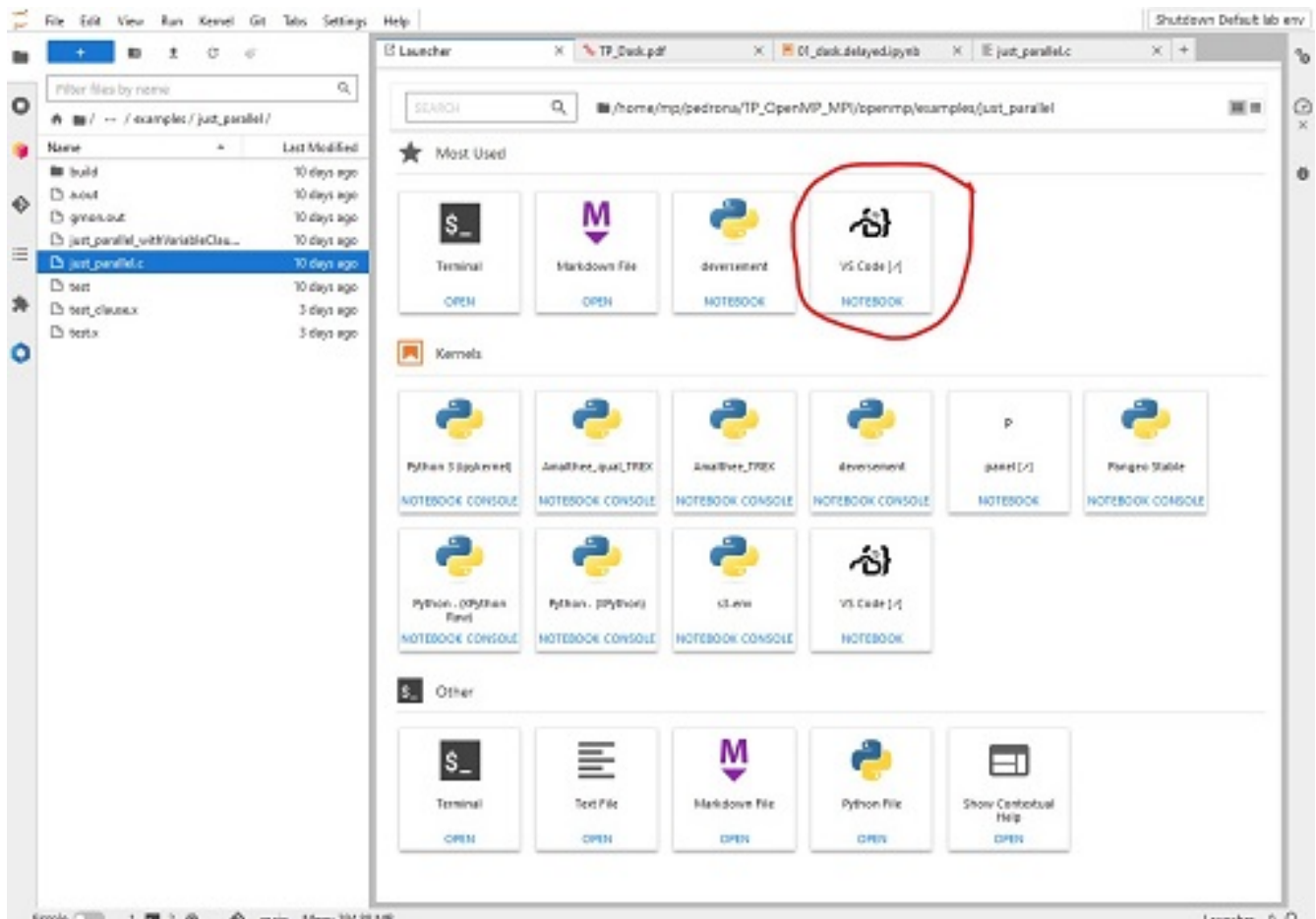


mais **attention, les modifications qui sont faites dans cette fenêtre ne sont pas sauvegardées !**

Pour éditer un fichier, vous pouvez utiliser vim ou nano ou tout autre éditeur dépourvu d'interface graphique depuis le terminal du Jupyter. Si vous ne connaissez aucun éditeur sans interface graphique, vous pouvez utiliser VS Code.

Utilisation de VS Code dans JupyterLab

VSCode est un éditeur proposé dans le Launcher.



VS Code va se lancer dans un autre onglet du navigateur et un mot de passe est demandé :

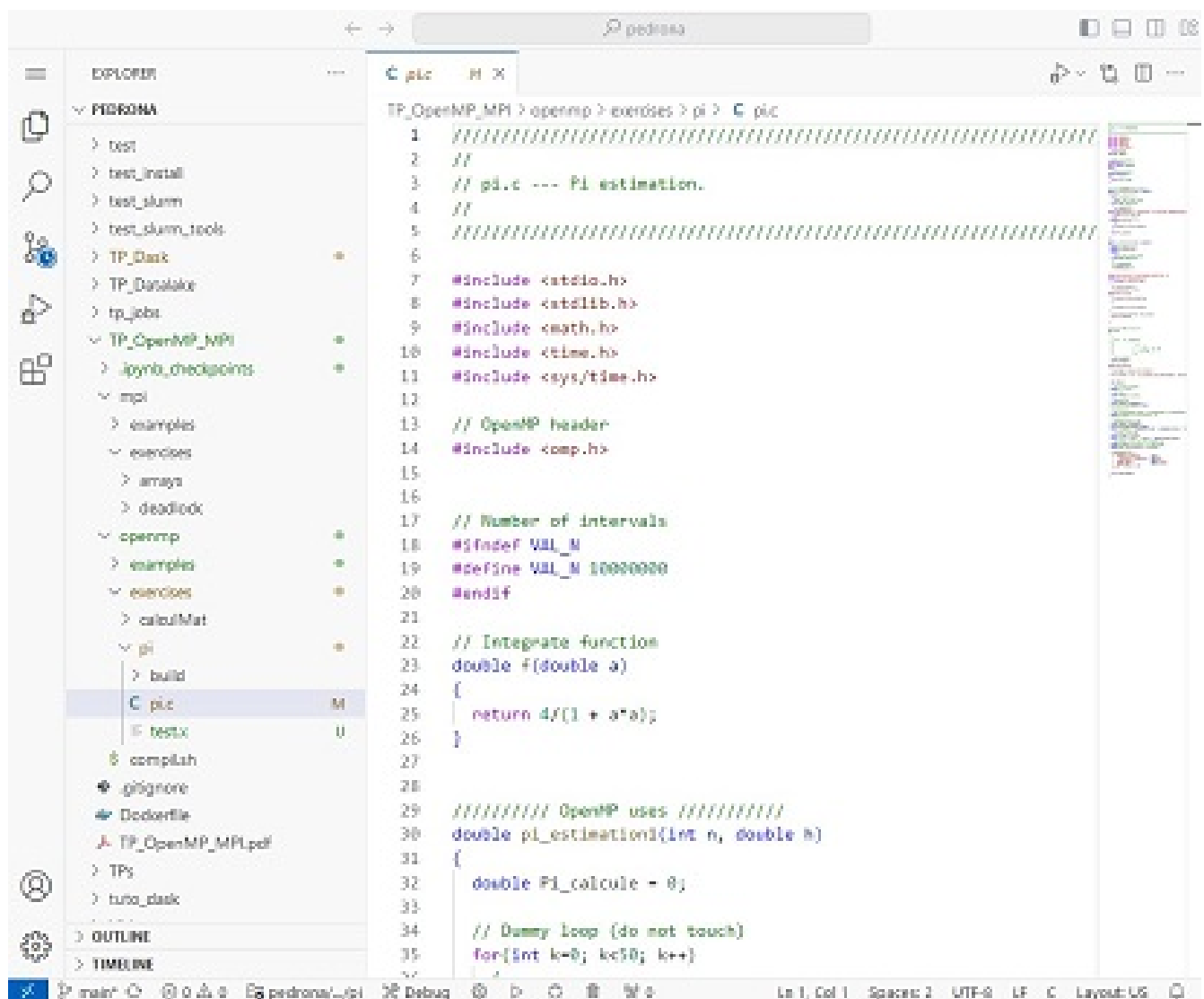
Welcome to code-server

Please log in below. Check the config file at `/home/.../.config/code-server/config.yaml` for the password.

Le mot de passe se trouve dans un fichier de configuration que vous pouvez retrouver grâce au terminal du Jupyter :

```
pedrona@tr X TP_Dask.pdf X 01_dask.dela X just_parallel X +
(/softs/rh8/conda-envs/pangeo_stable) [pedrona@trexvisu09 just_parallel]$
more /home/mp/pedrona/.config/code-server/config.yaml
bind-addr: 127.0.0.1:8080
auth: password
password: 06624516748160929c9112cb
cert: false
(/softs/rh8/conda-envs/pangeo_stable) [pedrona@trexvisu09 just_parallel]$
```

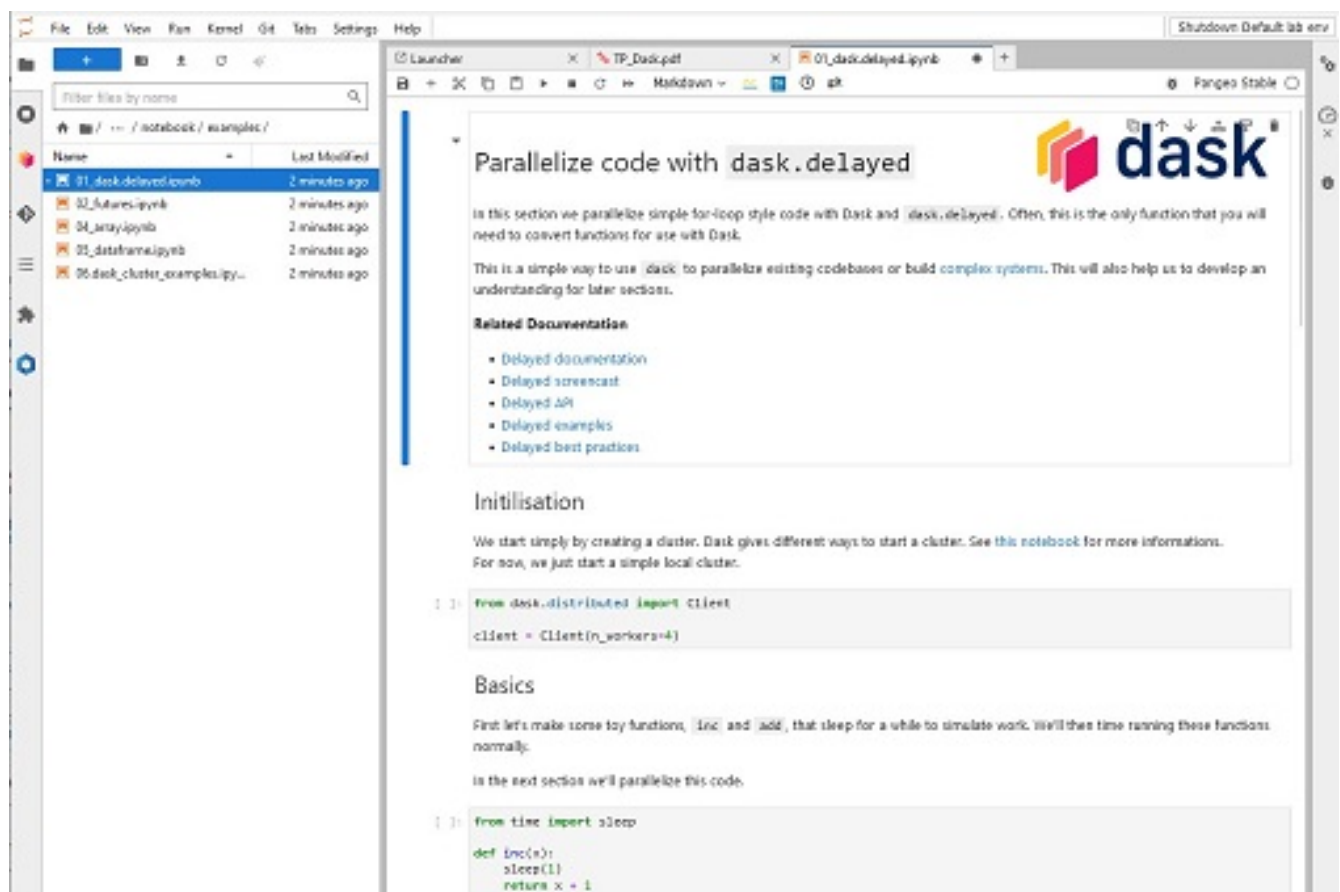
Vous pouvez alors éditer le fichier source *.c dans VS Code. VS Code est configuré en mode "autosaving". Les modifications seront sauvegardées automatiquement.



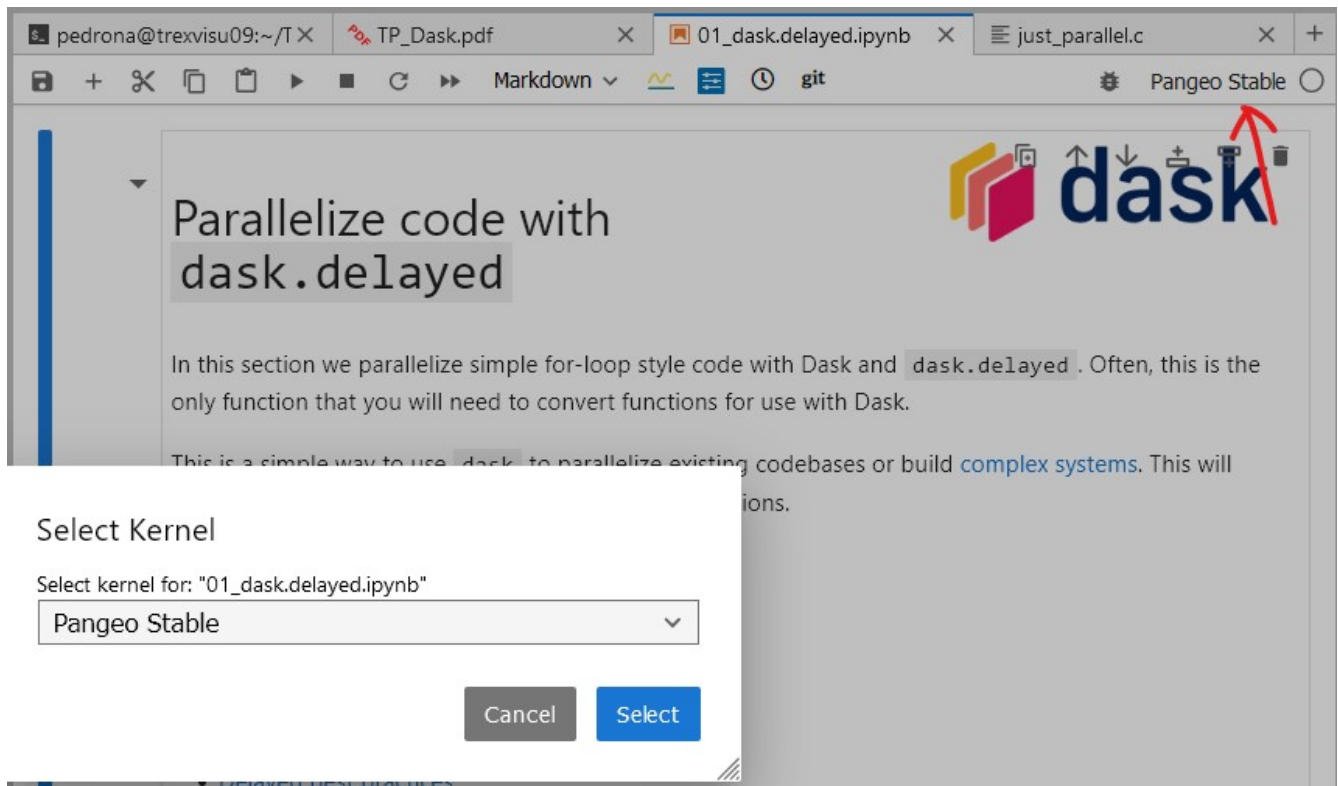
2 Utilisation de JupyterLab pour le 3ème TP (Dask)

2.1 Notebook Python

Un notebook Python (fichier avec une extension .ipynb) s'ouvre dans le Jupyterlab en cliquant sur le fichier. Un notebook Jupyter est un environnement interactif pour écrire et exécuter du code. Un notebook est capable d'exécuter du code dans une large gamme de langages. Cependant, chaque notebook est associé à un kernel. Les notebooks que nous allons utiliser sont associés à un kernel Python pour écrire et exécuter du code en Python.



Vous pouvez garder le kernel par défaut ou le changer en cliquant en haut à droite.



Les cellules permettent d'écrire et exécuter du code. Pour exécuter une cellule, utiliser "Shift-Enter" ou cliquer que le bouton "triangle" dans la barre d'outils en haut de la fenêtre.

Il y a un autre raccourci clavier : "Alt-Enter" exécute la cellule courant et en insert une autre sous la cellule courante. Vous pouvez utiliser le clic droit sur le notebook pour avoir accès à d'autres actions.

2.2 Gestion du kernel

Le code est exécuté dans un processus séparé appelé le kernel. Le kernel peut être interrompu ou redémarré en cliquant sur le bouton "carré" dans la barre d'outils. Si le kernel meurt, vous serez invité à le redémarrer.

2.3 Redémarrage des kernels

Le kernel conserve l'état des calculs d'un bloc-notes. Vous pouvez réinitialiser cet état en redémarrant le kernel. Cela se fait en cliquant sur la flèche qui tourne sur elle-même dans la barre d'outils.

Toutes les sorties sont affichées de manière asynchrone au fur et à mesure qu'elles sont générées dans le kernel.