



DATA
TERRA



ODATIS

Présentation activités VRE

Journée ODATIS - Océan Indien

Gwenaël CAËR, CNRS



contact@odatis-ocean.fr | www.odatis-ocean.fr

Qu'est ce qu'un VRE ?

Un **environnement virtuel de recherche** (VRE, de l'anglais *Virtual Research Environment*) est une **plateforme numérique intégrée** qui rassemble **les données, ressources et services** dont les chercheurs ont besoin pour mener un projet de recherche, souvent de façon collaborative et à distance.



Dans **un laboratoire**, un chercheur a accès à :

- des instruments / équipements
- des produits
- des ouvrages scientifiques
- des techniciens
- ...

Qu'est ce qu'un VRE ?

Un **environnement virtuel de recherche** (VRE, de l'anglais *Virtual Research Environment*) est une **plateforme numérique intégrée** qui rassemble **les données, ressources et services** dont les chercheurs ont besoin pour mener un projet de recherche, souvent de façon collaborative et à distance.



Dans **un VRE**, un chercheur a accès à :

- des logiciels / ressources de calcul
- des données
- de la documentation
- du support technique
- ...

Qu'est ce qu'un VRE ?

Dans le cadre d'ODATIS, les activités VRE consistent en la mise en place d'une **plateforme numérique intégrée** reposant sur 3 axes : **les données, les ressources et les services.**

Plateforme Numérique Intégrée (Jupyter)

Données

- Formats et protocoles
- Catalogues STAC

Ressources

- Calcul haute performance
- Environnements Python, Julia, R
- Outils intégrés : Dashboard, SIG, VNC

Services

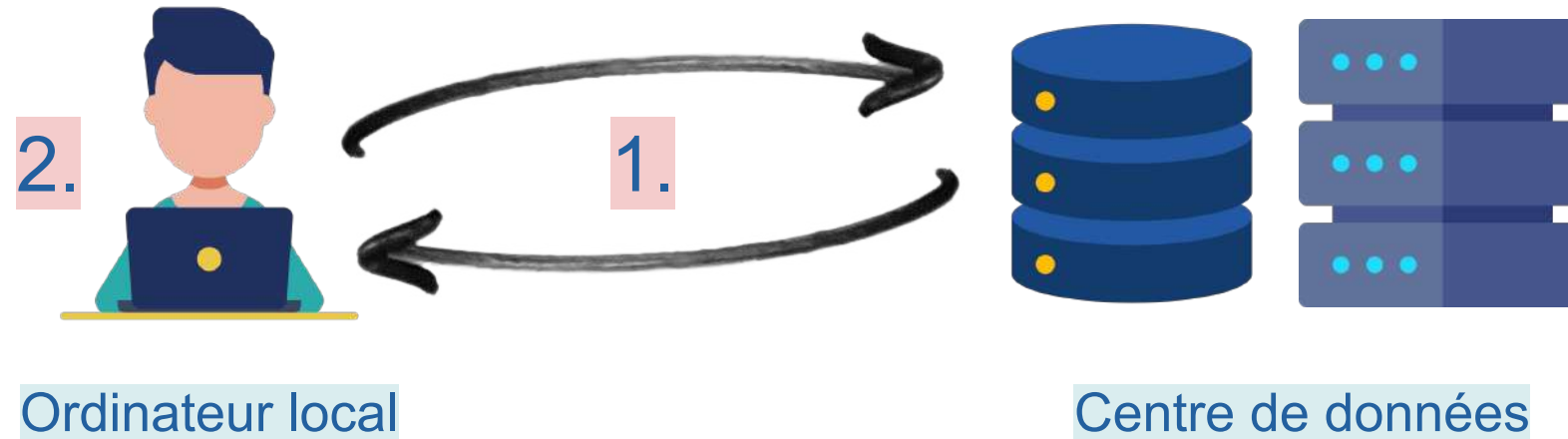
- Documentation ODATIS
- Cas d'usage scientifiques
- Cookbooks étape par étape
- Support technique

Pourquoi utiliser un VRE ?

1. Pour changer de paradigme

Pendant longtemps, pour analyser des données, il fallait :

1. Télécharger les données depuis le centre de données distant.
2. Analyser les données en local avec des ressources de calcul limitées.

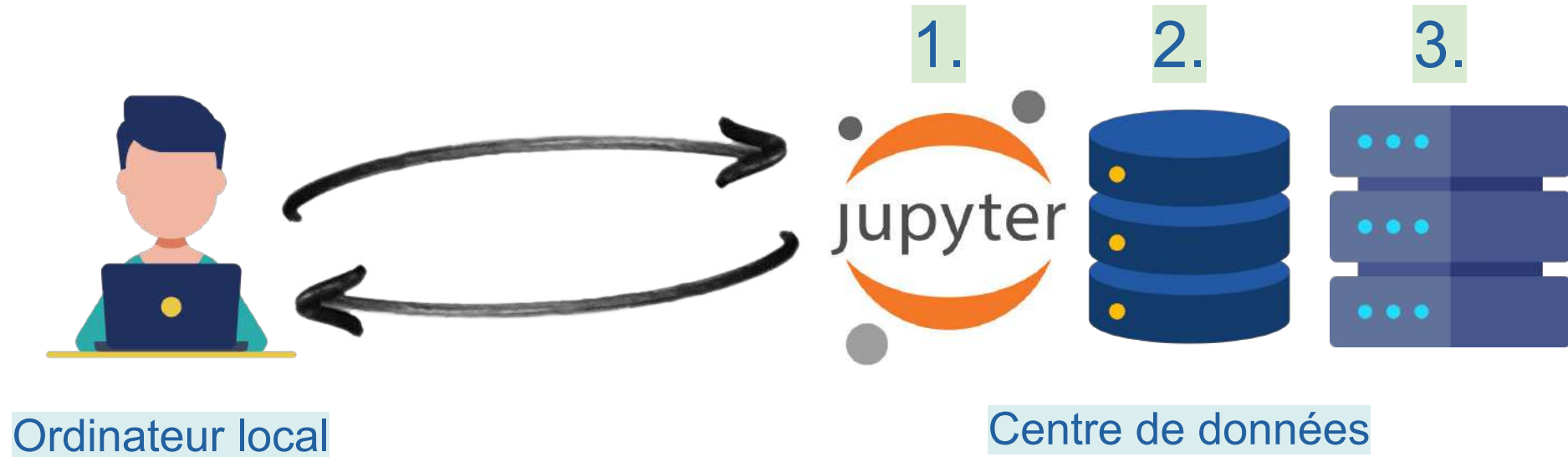


Pourquoi utiliser un VRE ?

1. Pour changer de paradigme

Avec un VRE, plus besoin de télécharger les données, tout se passe à distance comme si on était en local, il suffit de :

1. Accéder à la plateforme via internet
2. Ouvrir les données depuis les espaces de stockage
3. Lancer les traitements sur les ressources de calcul



Pourquoi utiliser un VRE ?

2. Pour simplifier l'accès et l'utilisation d'un cluster de calcul

Qu'est-ce que j'entends par là ? Voici un scénario dans lequel certains se reconnaîtront peut-être.

Vous êtes un chercheur / ingénieur, et vous avez besoin d'accéder à plus de ressources de calcul, vous demandez au service informatique de votre institut et il vous répond.

« Si tu as besoin de plus de ressources de calcul, tu peux lancer un job sur le cluster. Ce n'est pas compliqué, tu veux que je te montre ? »

Ça y est, quelqu'un est enfin prêt à vous montrer comment utiliser le cluster dont vous entendez parler depuis longtemps, et là...



Pourquoi utiliser un VRE ?

2. Pour simplifier l'accès et l'utilisation d'un cluster de calcul



« Alors tu commences par te connecter en SSH, ensuite t'arrives sur les nœuds de login, mais attention faut pas lancer de traitement sur les noeuds de login sinon tu risques de tout faire planter, donc faut lancer un job, pour ça tu définis tes ressources de calcul, quelle queue tu souhaites utiliser ? combien de CPU et de RAM il te faut ? Mais attention faut pas trop en demander parce que si tu réserves des ressources que tu n'utilises pas, ça pénalise les autres. Et si ça crashe, tu n'as qu'à aller

voir dans les logs. ... »

Pourquoi utiliser un VRE ?

2. Pour simplifier l'accès et l'utilisation d'un cluster de calcul

L'objectif d'un VRE est de simplifier l'accès et l'utilisation d'un cluster de calcul :

- en abaissant la barrière à l'entrée,
- en s'affranchissant du jargon technique,
- en masquant la complexité inutile pour l'utilisateur,
- et en proposant une interface ergonomique et intuitive.

1 Accéder à l'URL (et se connecter)

<https://jupyterhub.cnes.fr> (ou ifremer.fr)

connexion

2 Sélectionner les ressources (CPU, RAM, Durée, ...)

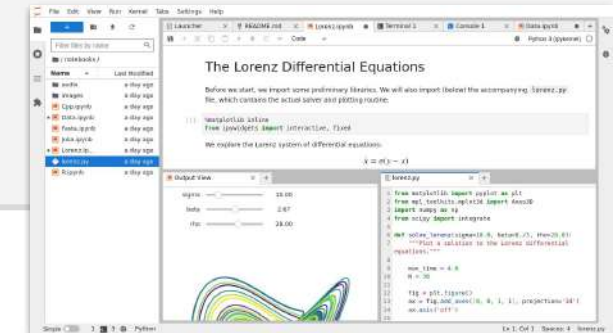
Sélection des ressources

1 coeur, 4gb, 8 heures

8 coeurs, 32gb, 8 heures

28 coeurs, 115gb, 8 heures

3 Commencer à utiliser le VRE (Données, Analyse, Visualisation, ...)



Plateforme numérique

Le projet Jupyter

Jupyter est un projet open-source pour le calcul interactif, construit autour de trois composants clés.

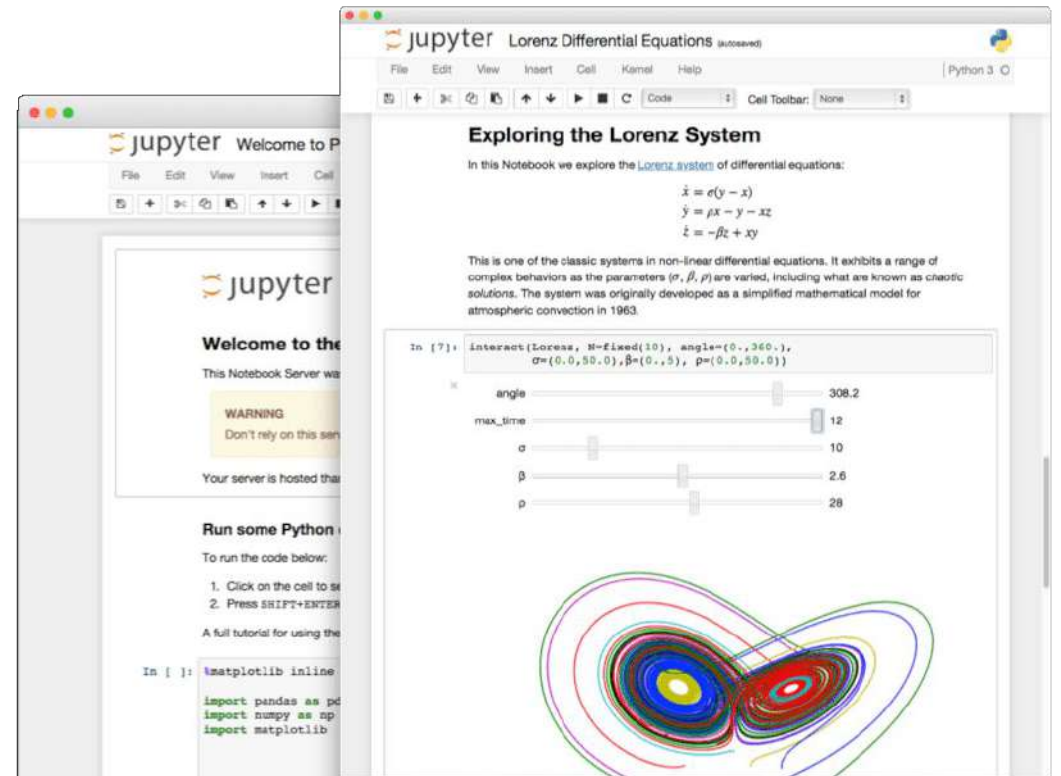
Plateforme numérique

Le projet Jupyter

Jupyter est un projet open-source pour le calcul interactif, construit autour de trois composants clés.

1. Jupyter Notebook

Le format de document qui concilie texte et code, pensé pour la conduite et la restitution des travaux de recherche.



Plateforme numérique

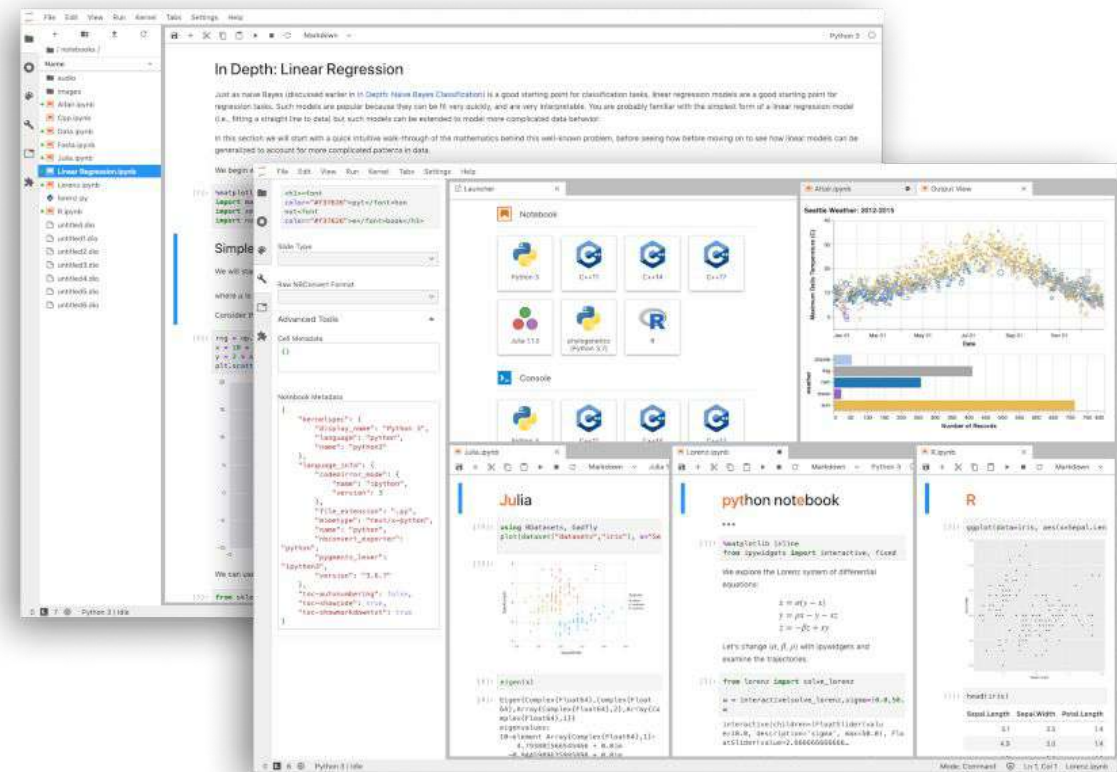
Le projet Jupyter

Jupyter est un projet open-source pour le calcul interactif, construit autour de trois composants clés.

1. Jupyter Notebook

2. JupyterLab

L'interface de travail qui donne accès aux outils, aux données et aux services de la plateforme, où s'exécutent les notebooks.



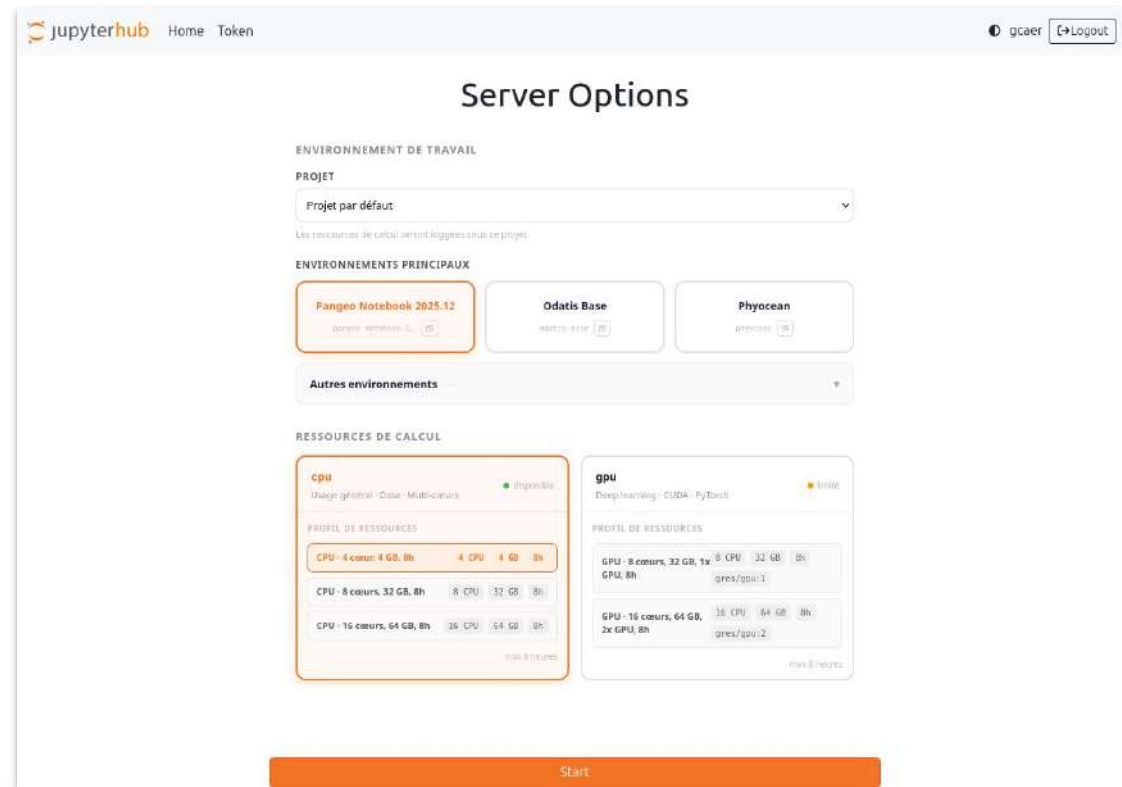
Plateforme numérique

Le projet Jupyter

Jupyter est un projet open-source pour le calcul interactif, construit autour de trois composants clés.

1. Jupyter Notebook
2. JupyterLab
3. JupyterHub

Le composant qui permet de demander des ressources de calcul et d'instancier un environnement JupyterLab pour chaque utilisateur.



Données

Formats et protocoles



Attention : Un panorama non exhaustif, basé sur mon expérience — il ne couvre pas nécessairement vos besoins ou cas d'usage.

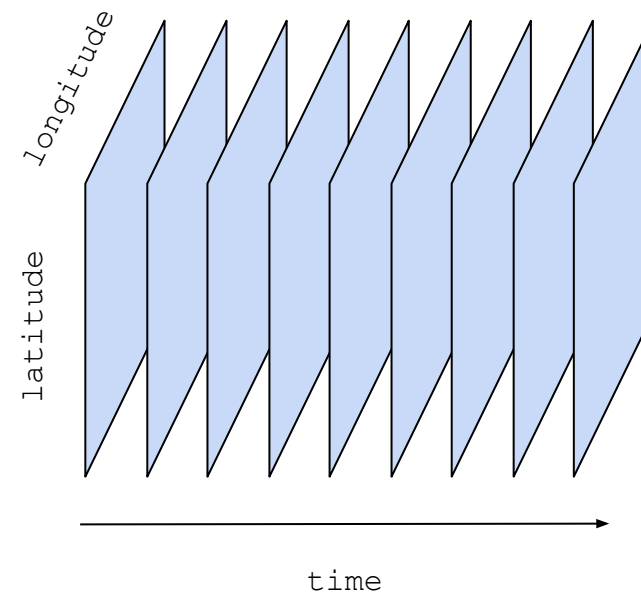
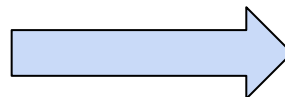
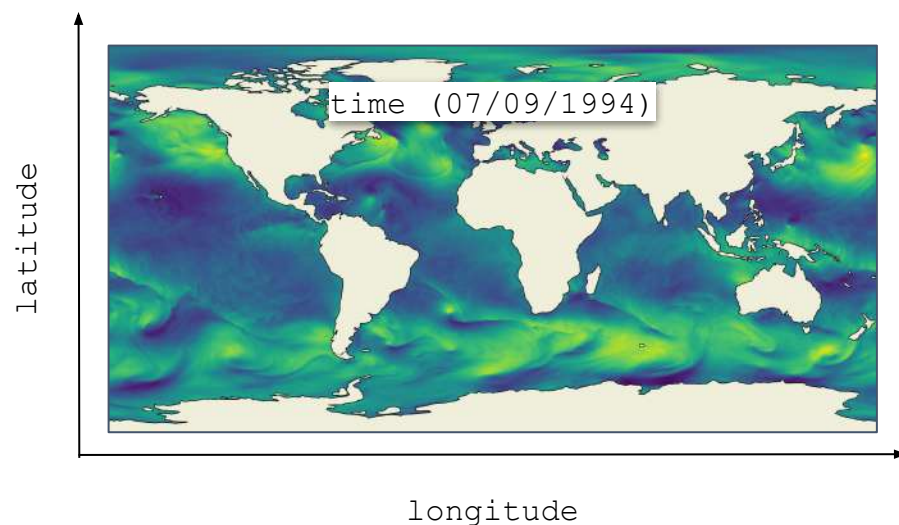


Données

Formats et protocoles

Historiquement, les données satellitaires et les données de modèles représentent des volumes très importants (plusieurs téraoctets).

Le format le plus couramment utilisé est le NetCDF, sous forme de fichiers journaliers à couverture globale, disponibles sur plusieurs décennies.



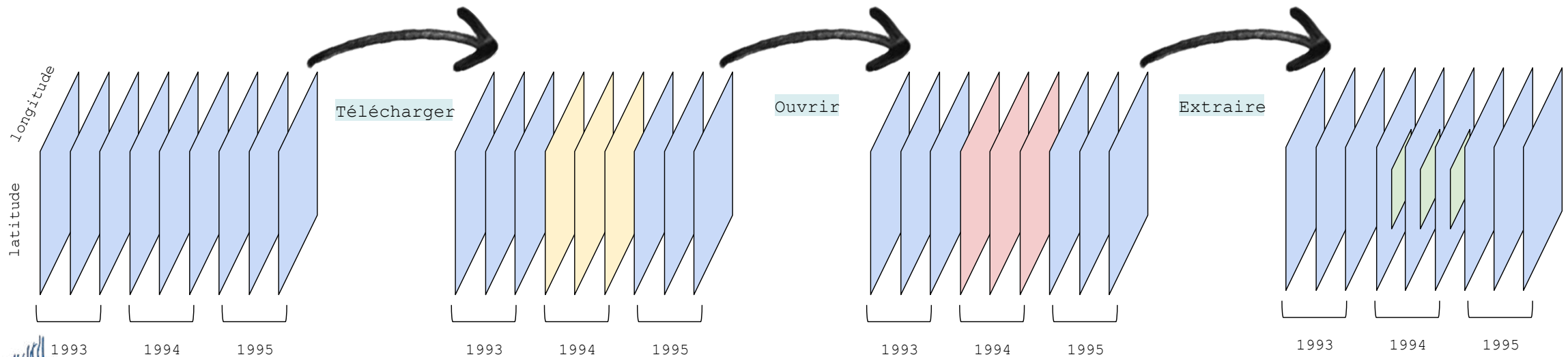
Données

Formats et protocoles



Si vous voulez accéder aux données de l'océan Indien sur l'année 1994

1. Télécharger l'ensemble des 365 fichiers (si vous êtes à distance).
2. Ouvrir chacun des 365 fichiers séparément.
3. Extraire la zone d'intérêt dans chacun des 365 fichiers.

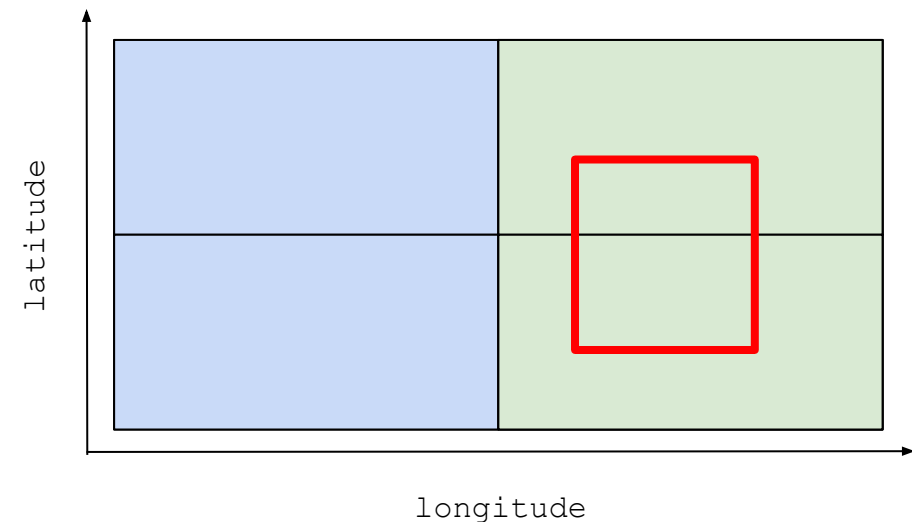
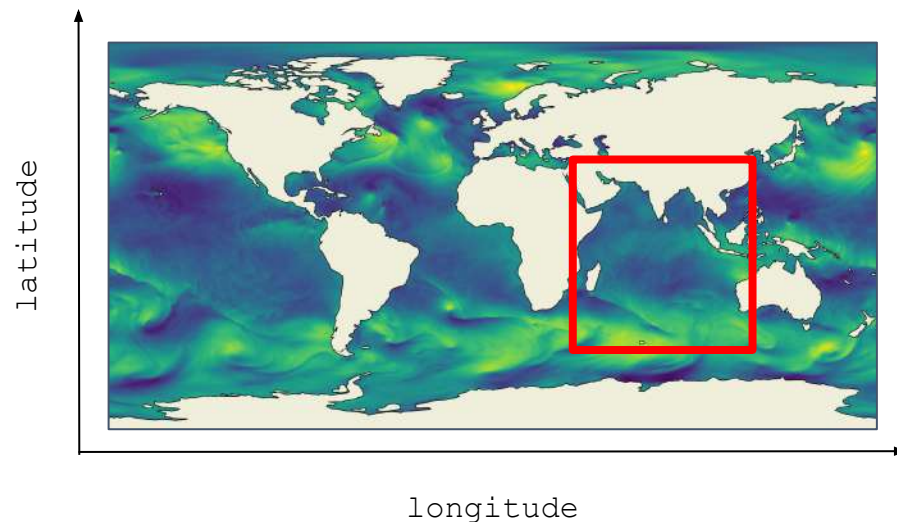


Données

Formats et protocoles

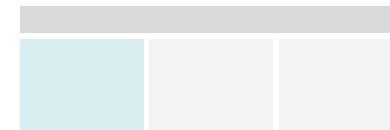


Le **format Zarr** a amélioré l'accès aux données grâce à un point d'accès unique et à un découpage en chunks. Cette approche a été renforcée par l'arrivée du **protocole S3**, qui permet désormais de télécharger uniquement les chunks nécessaires, sans avoir à récupérer l'intégralité du fichier.



Données

Formats et protocoles



Convertir en Zarr des jeux de données déjà en NetCDF reviendrait à **dupliquer des téraoctets de données**. C'est pour répondre à ce problème qu'est né **VirtualiZarr**, qui permet de **créer des Zarr virtuels à partir de fichiers NetCDF existants** — sans les dupliquer.

Une des **activités VRE** consiste à **générer des Zarr virtuels à partir des jeux de données existants au format NetCDF** (ERA5, produits satellitaires, etc.), afin d'offrir aux utilisateurs un accès rapide et simplifié aux données.

NB : L'Ifremer et le CNES sont en train d'implémenter le protocole S3 au sein de leur infrastructure afin de permettre un accès efficient aux données volumineuses.

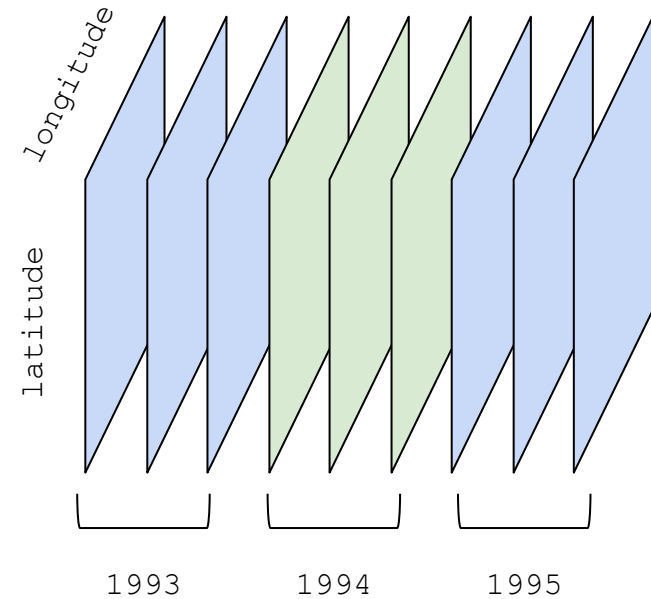
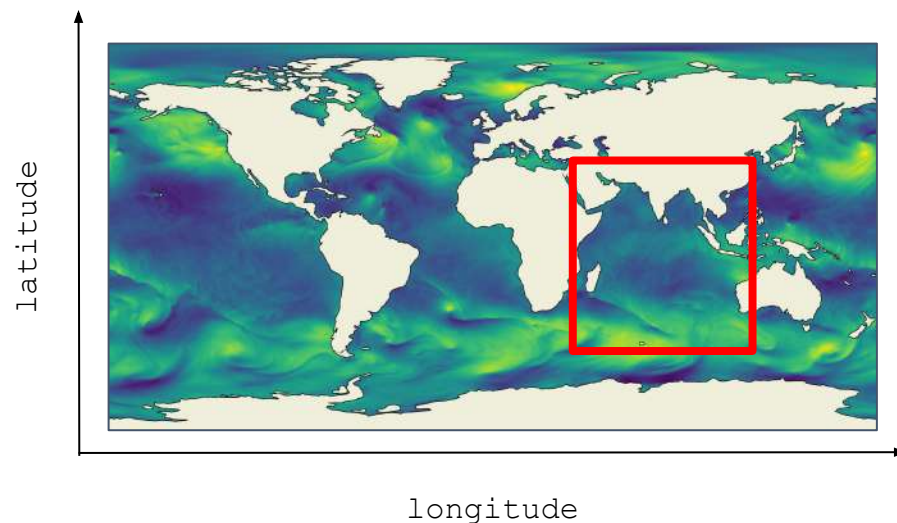


Les Données

Formats et protocoles

Si vous voulez accéder aux données de l'océan indien sur l'année 1994:

1. il est nécessaire de télécharger l'ensemble des 365 fichiers (si vous êtes à distance)
2. Il est nécessaire d'accéder à chacun des 365 fichiers séparément.
3. Il est nécessaire d'ouvrir en entier chaque fichier pour extraire la zone d'intérêt.

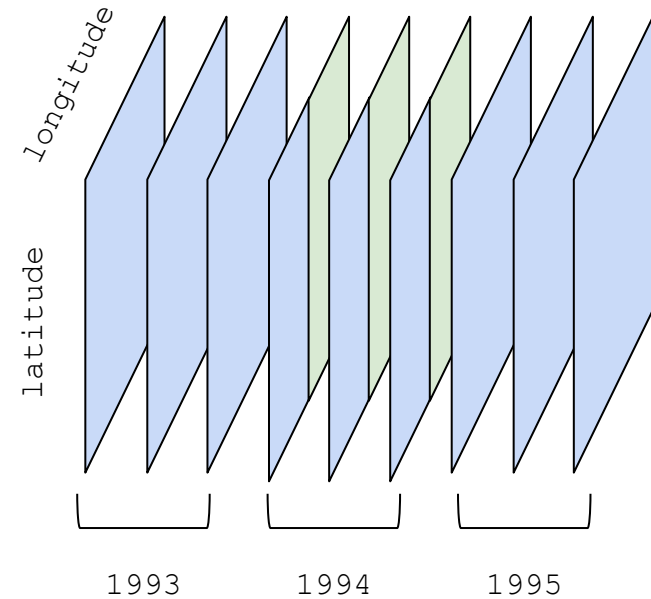
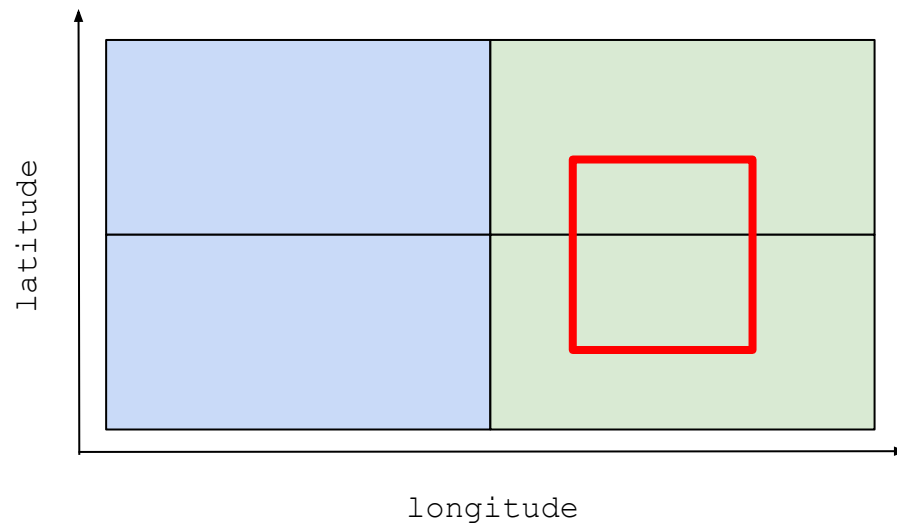


Les Données

Formats et protocoles



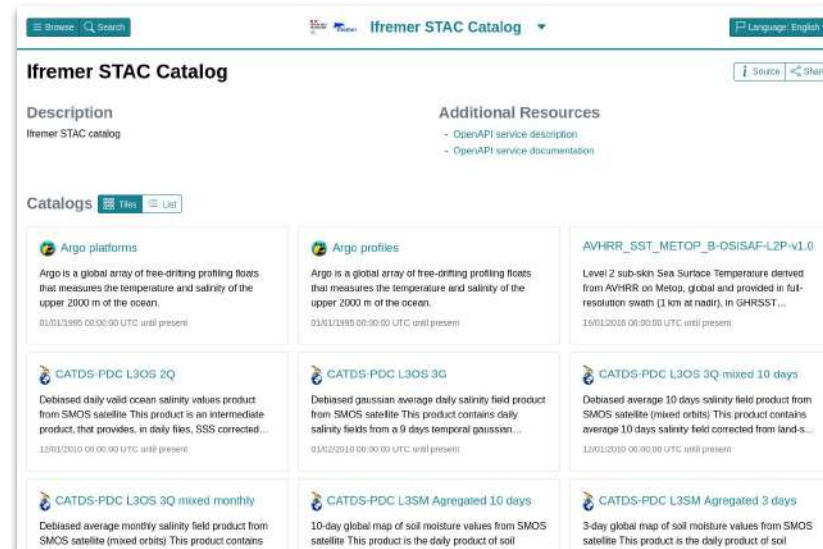
L'arrivée du chunking a permis d'améliorer l'accès aux données en segmentant la données en chunks.



Données

Catalogue STAC

STAC (SpatioTemporal Asset Catalog) est une spécification qui standardise la description et la découverte de données géospatiales, facilitant leur recherche et leur accès à travers différents catalogues.



NB : L'Ifremer et le CNES sont en train de mettre en place un catalogue STAC sur certains jeux de données.

Ressources

Les infrastructures HPC

Le pôle ODATIS met à disposition l'accès aux infrastructures de calcul du CNES (*TREX*) et de l'Ifremer (*DATARMOR*).



Cluster du CNES (TREX)



Cluster de l'Ifremer (DATARMOR)

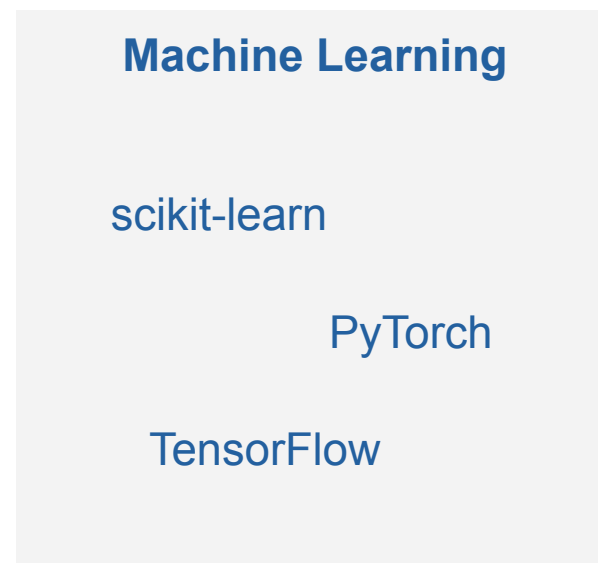
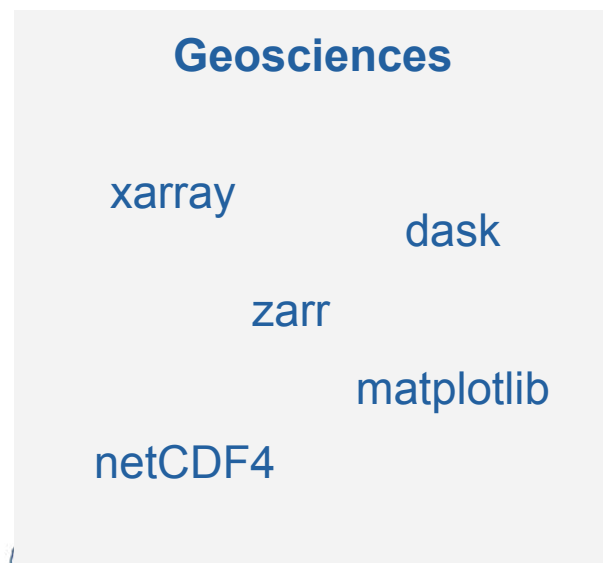
Ressources

Environnements Python, Julia, R



Les environnements regroupent, pour un langage donné (Python, Julia, R...), l'interpréteur et un ensemble cohérent de bibliothèques scientifiques prêtes à l'emploi — permettant à l'utilisateur de travailler immédiatement sans avoir à installer ou configurer quoi que ce soit.

1. Environnement standard



2. Environnement personnalisé



Ressources

Outils intégrés : Dask

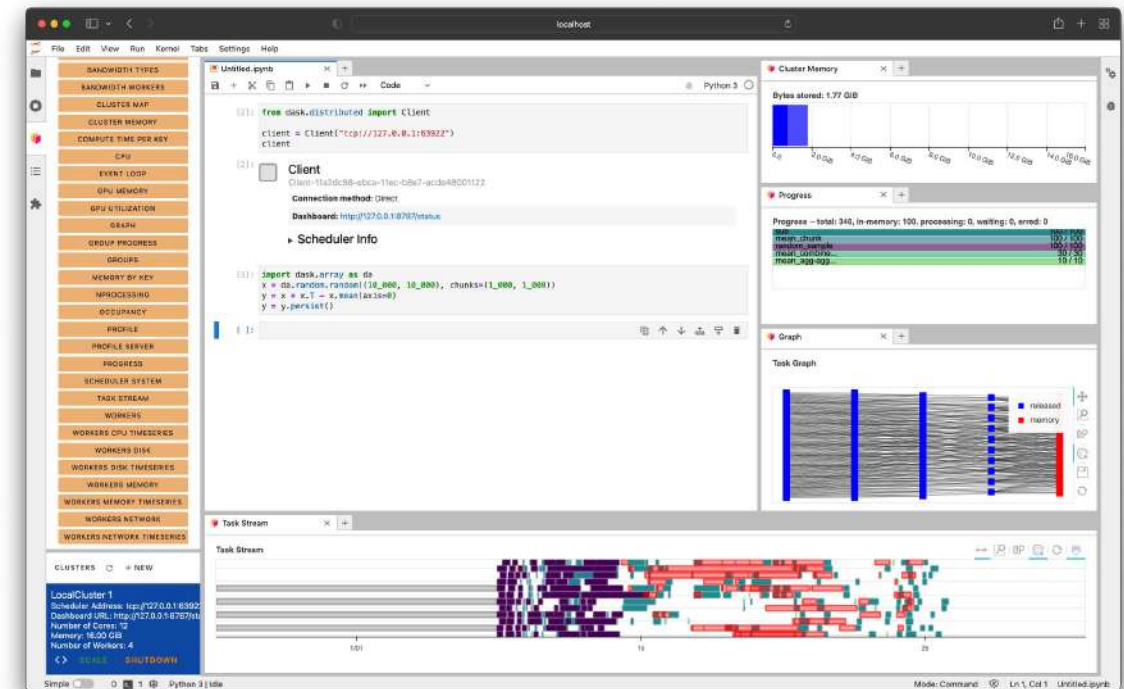
Dask est une bibliothèque Python qui permet de répartir un calcul sur l'ensemble des cœurs disponibles, afin de traiter efficacement des jeux de données volumineux.



```
from dask.distributed import Client, LocalCluster

cluster = LocalCluster(
    n_workers=5,          # nombre de workers
    threads_per_worker=2, # 5 x 2 = 10 threads (≈ 10 cœurs)
    memory_limit="16GB"   # 5 x 16GB = 80GB
)

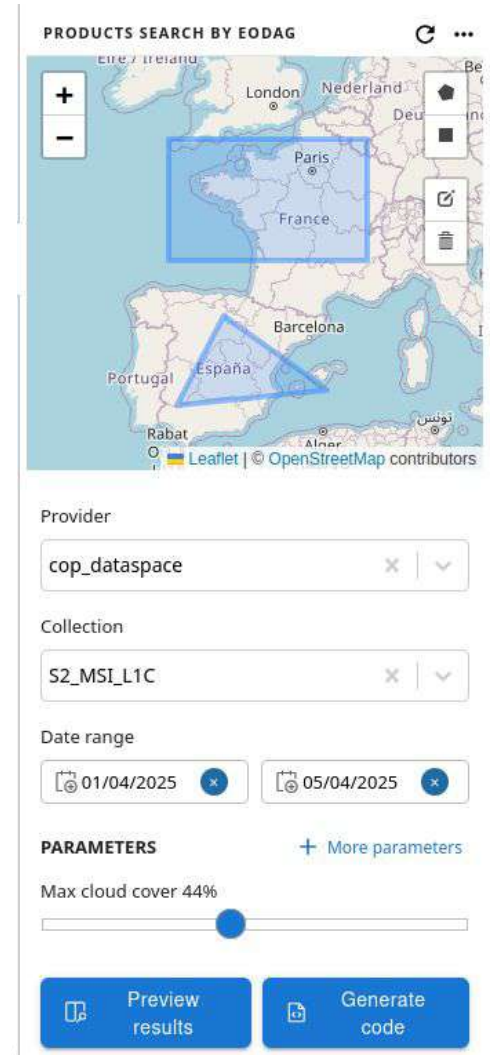
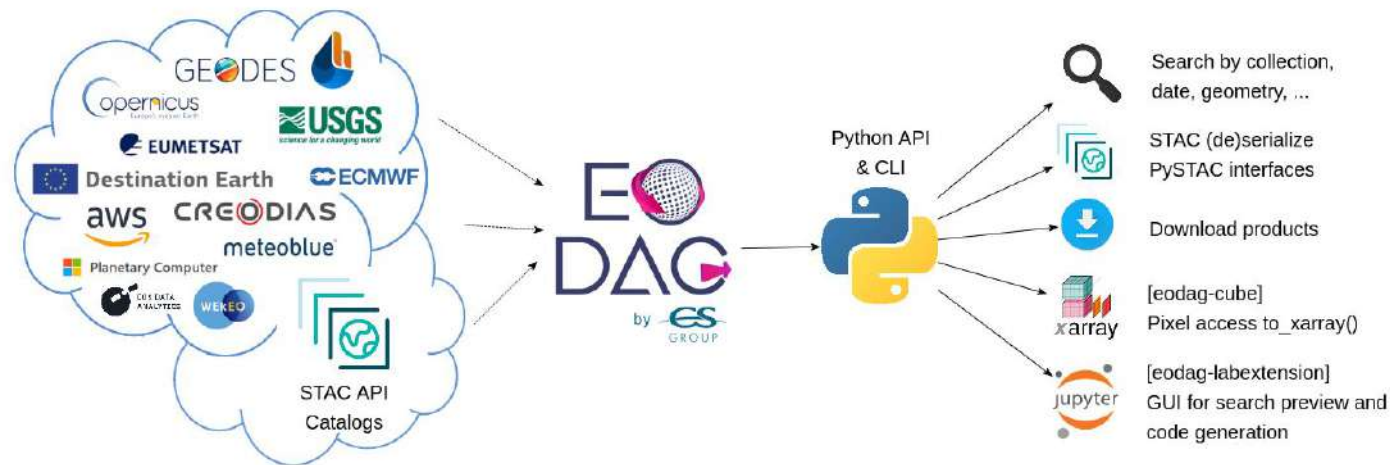
client = Client(cluster)
```



Ressources

Outils intégrés : EODAG

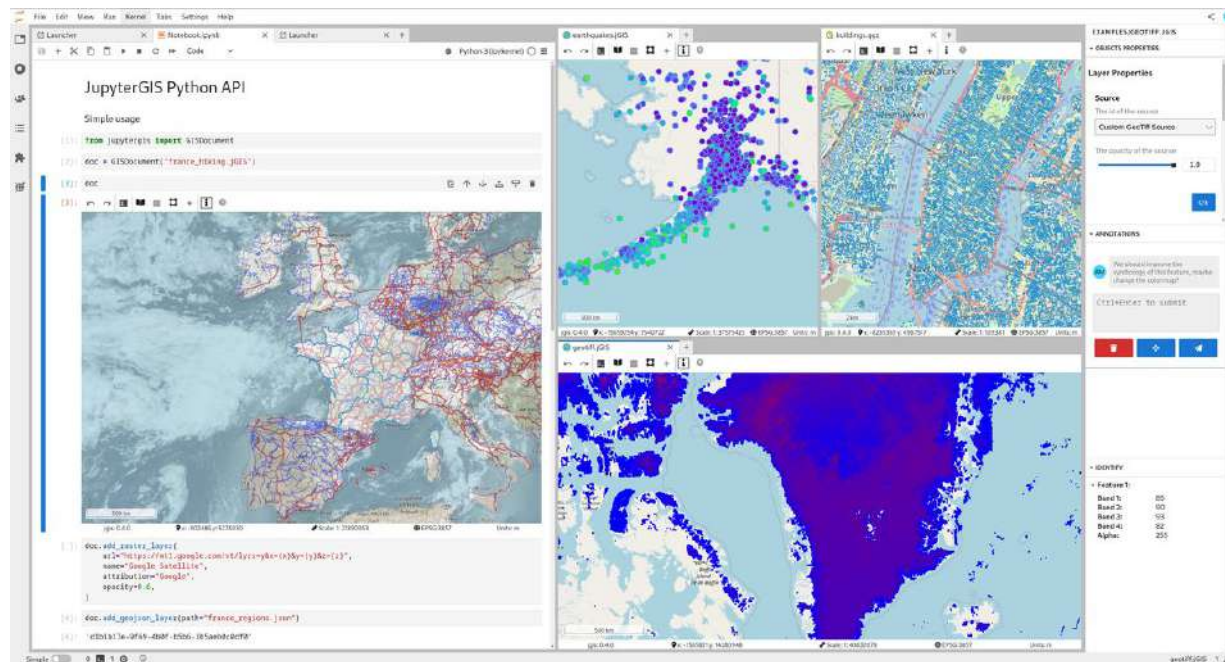
EODAG est une bibliothèque Python qui offre un point d'accès unique pour rechercher et télécharger des données d'observation de la Terre auprès de différents fournisseurs, notamment via une connexion aux catalogues STAC.



Ressources

Outils intégrés : JupyterGIS

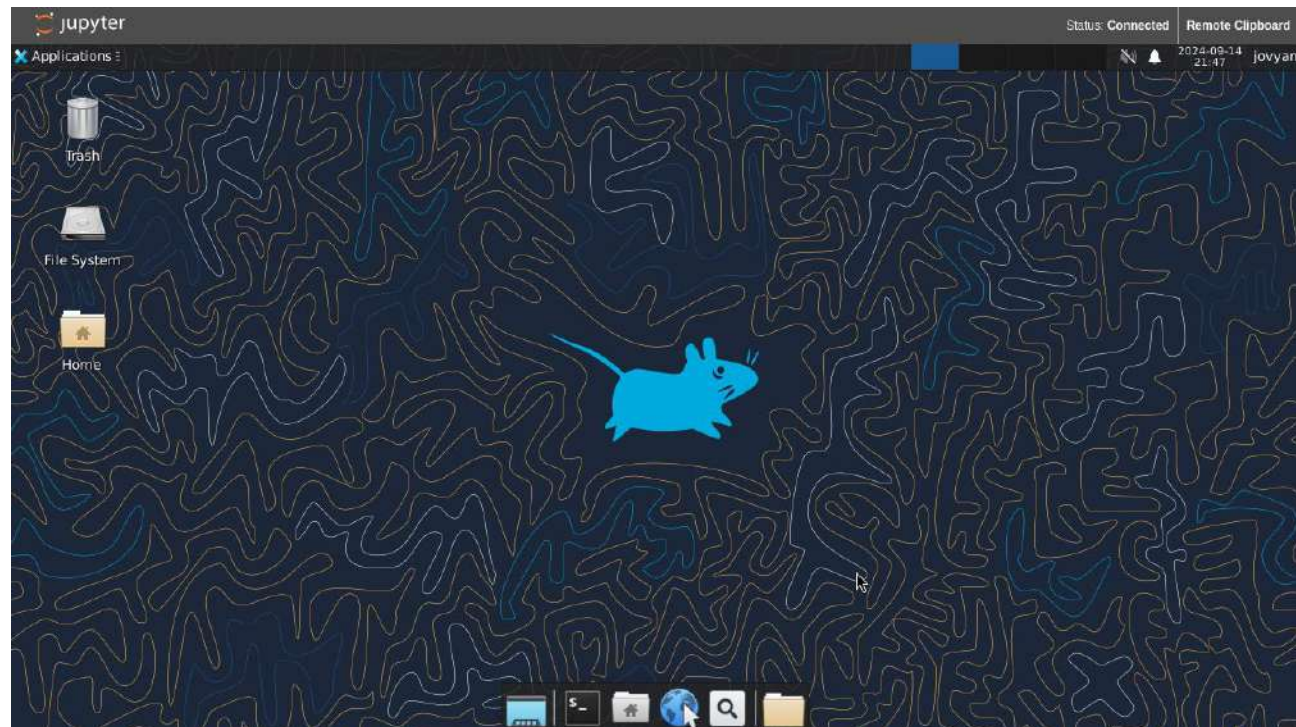
JupyterGIS est une extension JupyterLab qui offre un environnement SIG, permettant de visualiser, analyser et manipuler des données géospatiales (rasters, vecteurs, fichiers QGIS...) directement dans l'interface Jupyter.



Ressources

Outils intégrés : VNC

Desktop (via VNC) est un outil qui donne accès à un bureau virtuel directement dans Jupyter, permettant de lancer des applications desktop classiques telles que QGIS.

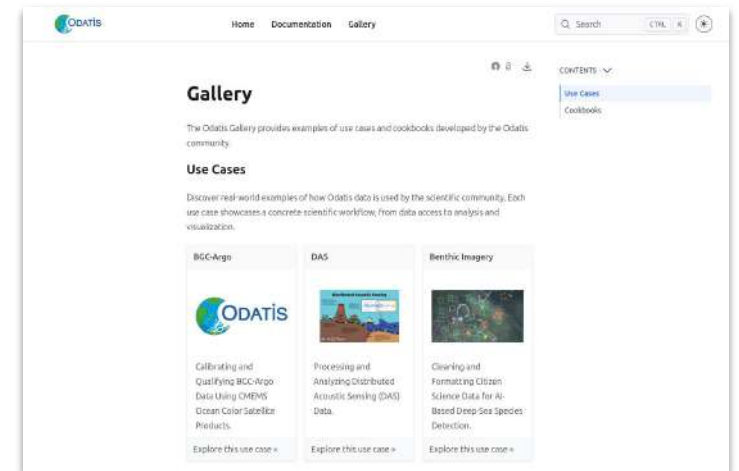
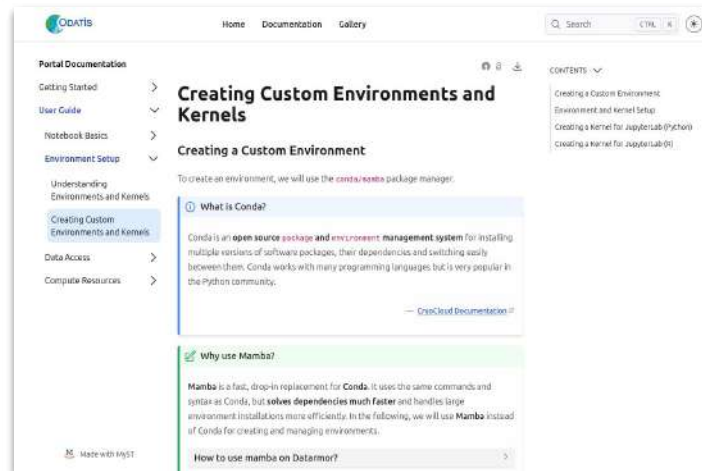
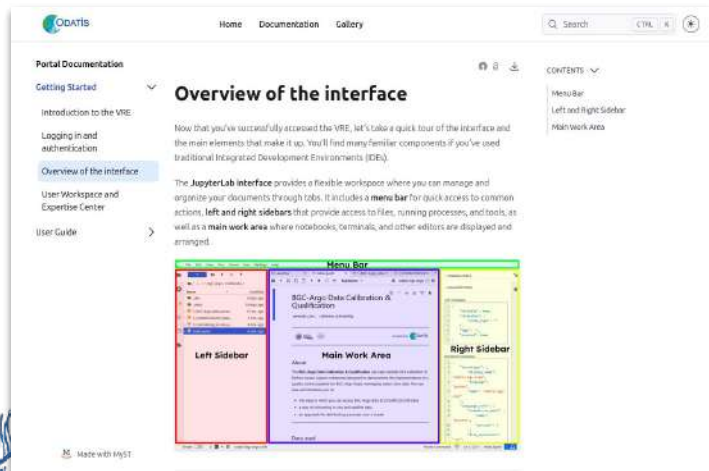


Services

Documentation ODATIS

La documentation ODATIS couvre trois besoins complémentaires :

- **Getting Started** : se connecter à la VRE, choisir ses ressources, découvrir l'interface
- **User Guide** : paralléliser ses calculs avec Dask, transférer ses données, créer son propre environnement...
- **Galerie** : cas d'usage et cookbooks

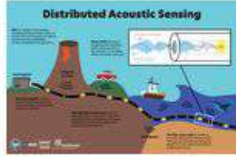


Services

Cas d'usage scientifiques

Dans le cadre d'ODATIS, trois **cas d'usage** principaux ont été mis en place. Ils s'appuient sur des exemples scientifiques concrets pour illustrer comment :

- travailler avec des données volumineuses,
- paralléliser des traitements,
- entraîner des modèles d'IA,
- ou encore colocaliser des données satellitaires et in situ.

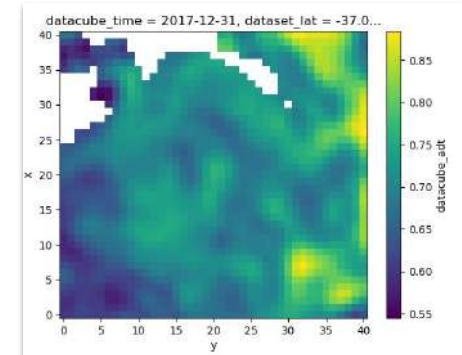
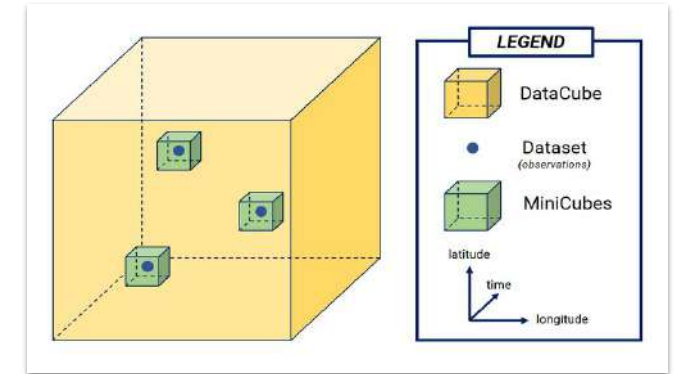
BGC-Argo	DAS	Benthic Imagery
		
Calibrating and Qualifying BGC-Argo Data Using CMEMS Ocean Color Satellite Products.	Processing and Analyzing Distributed Acoustic Sensing (DAS) Data.	Cleaning and Formatting Citizen Science Data for AI-Based Deep-Sea Species Detection.
Explore this use case »	Explore this use case »	Explore this use case »

Services

Cas d'usage scientifiques

Colocalisation des données in situ et satellitaires

Dans le cadre de nombreux travaux de recherche, une étape importante est la colocalisation des données. L'idée de ce cas d'usage est d'étudier comment utiliser des outils communs aux géosciences (Xarray, VirtualiZarr, Dask) pour optimiser la colocalisation d'un grand nombre de données.



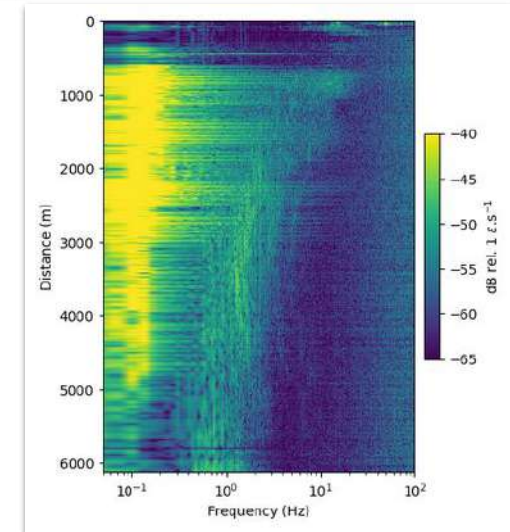
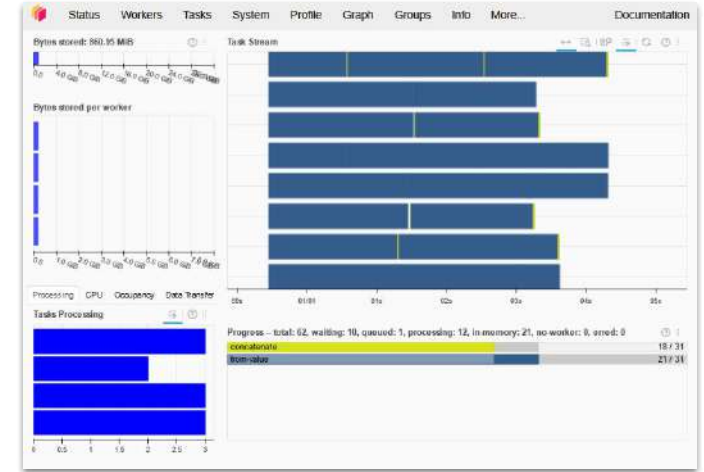
 PytCube

Services

Cas d'usage scientifiques

Traitement des données de fibre optique (DAS)

Dans le cadre d'une campagne menée en Sicile par une équipe de [Geo-Ocean](#), un volume très important de données a été acquis (environ 20 To). L'idée de ce cas d'usage est d'étudier comment optimiser la lecture, le traitement et la visualisation d'un tel volume de données.



Services

Cas d'usage scientifiques

Traitement des données d'imagerie benthique

Dans le cadre du projet [DeepSeaSpy](#) mené par le LEP, un grand nombre d'annotations d'espèces provenant de la science participative sont disponibles. L'idée de ce cas d'usage est d'étudier comment utiliser des données citoyennes pour entraîner un modèle d'IA, afin d'annoter de futures images de manière automatique.



Services

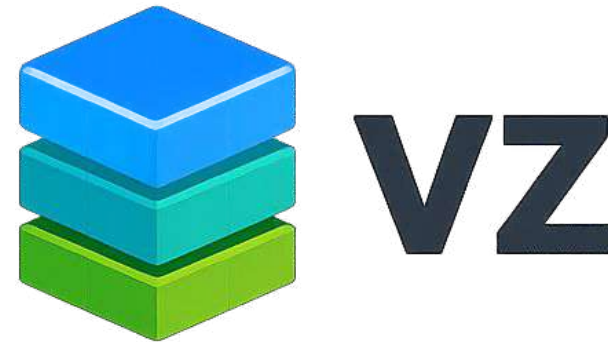
Cas d'usage scientifiques

Ces cas d'usage ont permis le développement de plusieurs librairies Python:

PytCube & Fast Vindex : Colocalisation de minicubes autour de points d'observation dans des datacubes multidimensionnels.

VZFlow : Workflow pour la création de Zarr Virtuels au format IceChunk à partir de readers configurés par dataset.

Daspal : Traitement de données DAS avec Xarray et Dask.



Services

Cookbooks étape par étape

Un **cookbook** est un **guide pratique et progressif** pour réaliser une tâche concrète. Dans le cadre d'ODATIS, on retrouve par exemple des cookbooks sur le **partage de code en package Python**, ou **la publication de recherches en site web statique avec MyST-MD**.

How to share your code as a reusable Python library?



A step-by-step guide to packaging, documenting, and distributing your Python code so others can install and reuse it.

[Explore this cookbook »](#)

How to publish your research as a website?



A step-by-step guide to turning your scientific notebooks and documentation into a structured, navigable, and shareable website.

[Explore this cookbook »](#)

How to share your code as a reusable Python library?

1. Git initialization
2. Development environment
3. Python package
4. Tests
5. Documentation
6. CI/CD with GitLab
7. Versioning
8. Publishing
9. Evolving the package
Glossary & Index

How to share your code as a reusable Python library?

Prerequisites
Example package
Share your feedback
Sources and acknowledgments

As a scientist, you have probably written code that works — a collection of utility functions, a data processing pipeline, or a simulation tool. But sharing it with colleagues, making it installable on any machine, and keeping it maintainable over time is a different challenge.

This cookbook walks you through the **complete workflow for packaging Python code properly**, from an empty folder to a versioned, documented, and automatically tested library deployed on GitLab.

Note

Not every chapter is required:

- Essential to create a package
- Recommended for good practices
- Advanced to automate and scale.

Chapter: 1. Git initialization
Level: Recommended

How to share your code as a reusable Python library?

1. Git initialization
2. Development environment
3. Python package
4. Tests
5. Documentation
6. CI/CD with GitLab
7. Versioning
8. Publishing
9. Evolving the package
Glossary & Index

3. Python package

Essential

This chapter is required to create a functional and installable Python package.

This chapter turns your Python code into an installable package. By the end, anyone will be able to install your library with a single `pip install` command.

Project structure

A Python package follows a specific directory layout. Here is what the final structure looks like for `cube3d`:

```
cube3d/
├── src/
│   ├── cube3d/
│   │   ├── __init__.py
│   │   └── core.py
│   └── LICENSE
├── README.md
└── pyproject.toml
```

Why put the code inside `src/`?

Placing your package under `src/` prevents a subtle but common bug: without it, Python can accidentally import your source code directly from the project root instead of the installed version — making tests pass locally but fail for your users. The `src/`

Services

Support technique



Le **support technique** se concentre principalement sur la maintenance et l'évolution des images Docker et des environnements de la VRE, ainsi que sur l'accompagnement dans la mise en place de cas d'usage.



Le VRE ODATIS

Une plateforme numérique intégrée

Le VRE ODATIS est le résultat de l'intégration de chacune de ses briques.

