

# PLMshift, NoteBooks et GPU

Philippe Depouilly  
Damien Ferney

October 2021



# Offre de services sur PLMshift

## Rappels au sujet de PLMshift

- ▶ Plateforme en tant que Service (PaaS) orientée utilisateurs
- ▶ Les membres de la communauté peuvent déployer des services web (application) et consommer de la ressource CPU, Mémoire, Disque et Réseau
- ▶ Ne nécessite pas l'instanciation d'une VM
- ▶ Dispense de l'administration d'un OS
- ▶ Permet de multiplier les instances (production, développement, tests)
- ▶ Fonctionne de manière déclarative :
  - ▶ on définit des ressources
  - ▶ le système les met à disposition
  - ▶ l'intelligence est dans une image de conteneur



# Déployer une application

Une application est un service accessible via le WEB (HTTPS) qui utilise de la CPU et de la RAM.

Cette application va pouvoir obtenir un FQDN et un certificat HTTPS (LetsEncrypt)

Trois méthodes pour déployer une application

- ▶ via le catalogue
- ▶ via un fichier déclaratif (yaml ou json)
- ▶ via une recette Helm3



# Déployer via la catalogue

Assent d'utiliser PLMshift, merci de consulter les CGU

okd

Filoulou Depou

Project: All Projects

### Topology

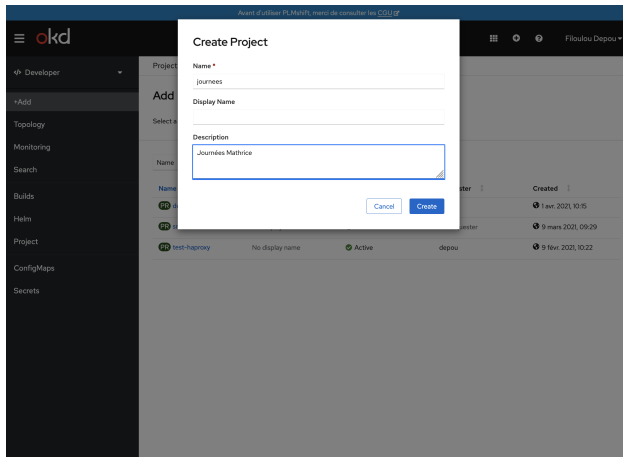
Select a Project to view the topology or [create a Project](#).

Name Search by name...

Name	Display name	Status	Requester	Created
depou-che	No display name	Active	depou	1 avr. 2021, 10:15
smf2020	No display name	Active	No requester	9 mars 2021, 09:29
test-haproxy	No display name	Active	depou	9 févr. 2021, 10:22



# Déployer via la catalogue



# Déployer via la catalogue

Avant d'utiliser PLMshift, merci de consulter les CGU [cf](#)

okd

Project: journees Application: all applications

### Add

Select a way to create an Application, component or service from one of the options.

- Quick Starts**
  - Get started with Spring
  - Monitor your sample application
  - Get started with Quarkus using a Helm Chart
  - [View all Quick Starts](#)
- Samples**
  - Create an Application from a code sample
- From Git**
  - Import code from your Git repository to be built and deployed
- From Devfile**
  - Import your Devfile from your Git repository to be built and deployed
- Container Image**
  - Deploy an existing Image from an Image registry or Image stream tag
- From Dockerfile**
  - Import your Dockerfile from your Git repository to be built and deployed
- YAML**
  - Create resources from their YAML or JSON definitions
- From Catalog**
  - Browse the catalog to discover, deploy and connect to services
- Database**
  - Browse the catalog to discover database services to add to your Application



# Déployer via la catalogue

Avant d'utiliser PLMshift, merci de consulter les CGU

okd

Project: journees

### Developer Catalog

Add shared applications, services, event sources, or source-to-image builders to your Project from the developer catalog. Cluster administrators can customize the content made available in the catalog.

All Items

Search: wordpress X A-Z 3 Items

- WordPress (Classic / Repository)  
Creates a WordPress installation with separate MySQL database instance. Plugins, themes and...
- WordPress (Classic)  
Creates a WordPress installation with separate MySQL database instance. Requires that two...
- WordPress (Kit CNRS)  
Creates a WordPress installation with separate MySQL database instance. Plugins, themes and...



# Déployer via la catalogue

Assistant d'utiliser PLMshift, merci de consulter les CGU

okd

Filoufou Depou ▾

Developer ▾

- +Add
- Topology
- Monitoring
- Search
- Builds
- Helm
- Project
- ConfigMaps
- Secrets

### Instantiate Template

**Namespace \***  
journees

**APPLICATION\_NAME \***  
my-wordpress-site  
The name of the WordPress instance.

**WORDPRESS\_VOLUME\_SIZE \***  
1Gi  
Size of the persistent volume for Wordpress.

**WORDPRESS\_VOLUME\_TYPE \***  
ReadWriteOnce  
Type of the persistent volume for Wordpress.

**VOLUME\_CLASS**  
ceph-block  
Storage class for persistent storage (leave for default)

**WORDPRESS\_DEPLOYMENT\_STRATEGY \***  
Recreate  
Type of the deployment strategy for Wordpress.

**WORDPRESS\_MEMORY\_LIMIT \***  
512Mi  
Amount of memory available to Wordpress.

**DATABASE\_VOLUME\_SIZE \***  
1Gi  
Size of the persistent volume for the database.

### WordPress (Classic)

QUICKSTART PHP WORDPRESS

Creates a WordPress installation with separate MySQL database instance. Requires that two persistent volumes be available. If a ReadWriteMany persistent volume type is available and used, WordPress can be scaled to multiple replicas and the deployment strategy switched to Rolling to permit rolling deployments on restarts.

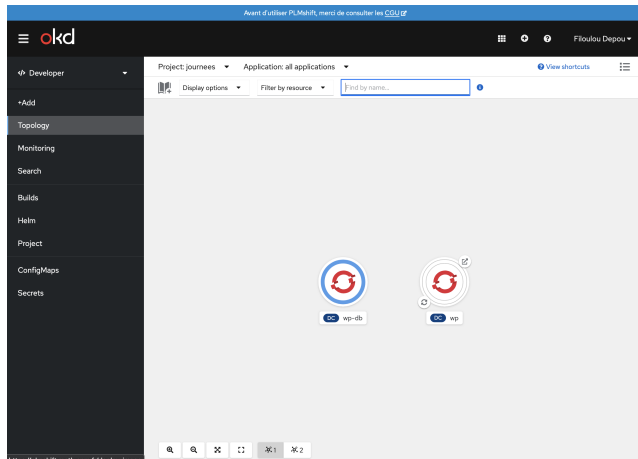
The following resources will be created

- BuildConfig
- DeploymentConfig
- ImageStream
- PersistentVolumeClaim
- Route
- Service





# Déployer via la catalogue

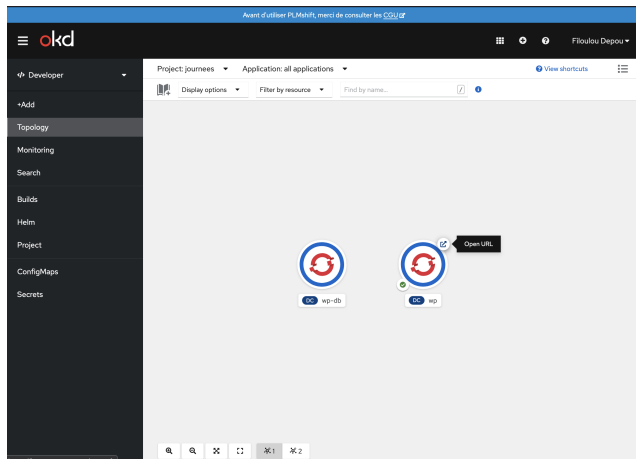


# Déployer via la catalogue

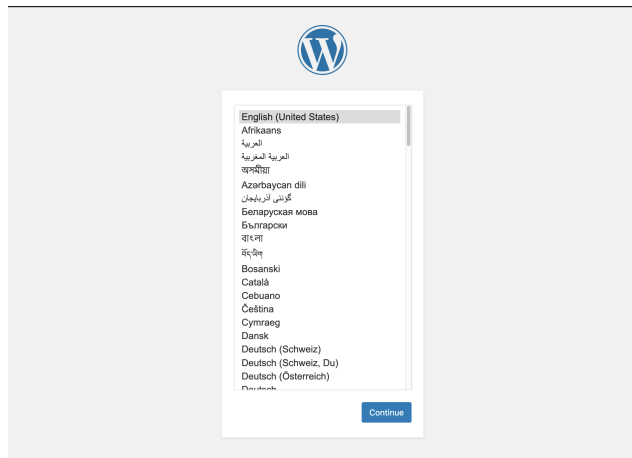
The screenshot shows the OKD web console interface. At the top, there is a navigation bar with the OKD logo and a user profile 'Filoulu Depou'. Below this is a sidebar with navigation options: Developer, +Add, Topology, Monitoring, Search, Builds, Helm, Project, ConfigMaps, and Secrets. The main area displays the 'Project: journeys' and 'Application: all applications'. A search bar is present with the text 'Find by name...'. The central part of the screen shows a topology view with two circular icons representing resources: 'dc wp-db' and 'dc wp'. The 'dc wp' resource is selected, and a detailed view panel is open on the right. This panel has tabs for 'Details', 'Resources', and 'Monitoring'. The 'Resources' tab is active, showing sections for 'Pods' (No Pods found for this resource), 'Builds' (one build 'wp' is running), 'Services' (one service 'wp' is running), and 'Routes' (one route 'wp' is available at 'https://journees.apps.math.cnrs.fr').



# Déployer via la catalogue



# Déployer via la catalogue



# Déployer en ligne de commande

OKD propose un interpréteur en ligne de commande **oc** qui est une extension de **kubectl**

```
$ oc login --token=sha256~pZ... --server=https://api.math.cnrs.fr:6443

$ oc new-project wp-de-depouill

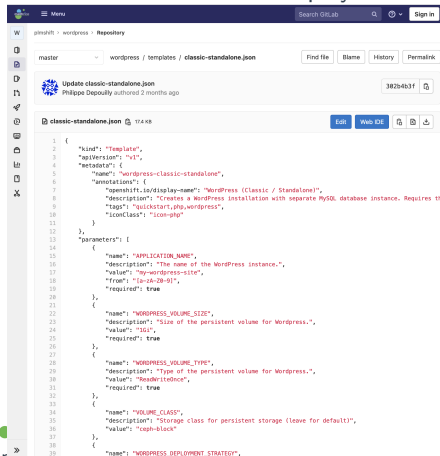
$ oc get templates -n openshift |grep wordpress
wordpress-classic-repository Creates a WordPress installation with separate MySQL database instance. Plugi... 15 (2)
wordpress-classic-standalone Creates a WordPress installation with separate MySQL database instance. Requi... 13 (1)
wordpress-kit-cnrs Creates a WordPress installation with separate MySQL database instance. Plugi... 14 (3 generated)

$ oc new-app wordpress-classic-standalone -p APPLICATION_NAME=mon_wordpress
```



# Déployer via un fichier

Sur <https://plmlab.math.cnrs.fr/plmshift>, vous pouvez trouver un ensemble de recettes déployables sur PLMshift



The screenshot shows the Oklab repository interface for the file 'classic-standalone.json'. The file content is a JSON configuration for a WordPress deployment template. The configuration includes metadata, annotations, and parameters for application name, volume size, volume type, and volume class.

```
1 {
2   "kind": "Template",
3   "apiVersion": "v1",
4   "metadata": {
5     "name": "wordpress-classic-standalone",
6     "annotations": {
7       "openshift.io/display-name": "WordPress (Classic / Standalone)",
8       "description": "Creates a WordPress installation with separate MySQL database instance. Requires t",
9       "tags": "quickstarts,php,wordpress",
10      "iconClass": "icon-php"
11    }
12  },
13  "parameters": [
14    {
15      "name": "APPLICATION_NAME",
16      "description": "The name of the WordPress instance.",
17      "value": "my-wordpress-site",
18      "from": "from-00-01",
19      "required": true
20    },
21    {
22      "name": "WORDPRESS_VOLUME_SIZE",
23      "description": "Size of the persistent volume for WordPress.",
24      "value": "1Gi",
25      "required": true
26    },
27    {
28      "name": "WORDPRESS_VOLUME_TYPE",
29      "description": "Type of the persistent volume for WordPress.",
30      "value": "ReadWriteOnce",
31      "required": true
32    },
33    {
34      "name": "VOLUME_CLASS",
35      "description": "Storage class for persistent storage (leave for default)",
36      "value": "ceph-block"
37    }
38  ],
39  "name": "WORDPRESS_DEPLOYMENT_STRATEGY",
```



# Déployer via un fichier

```
$ oc process -p APPLICATION_NAME=mon_wordpress \  
  https://plmlab.math.cnrs.fr/plmshift/wordpress/-/raw/master/templates/classic-standalone.json \  
  | oc create -f -
```

# Déployer via Helm3

```
$ helm repo add stable https://kubernetes-charts.storage.googleapis.com/
"stable" has been added to your repositories

$ helm repo update

$ helm install example-mysql stable/mysql

$ helm list
NAME      namespace    revision    updated      status    chart    app    version
example-mysql  mysql        1           2019-12-05  15:06:51.379134163 -0500 EST  deployed  mysql-1.5.0  5.7.27
```





# Anatomie d'une application

Il est bon d'avoir en tête les choses suivantes :

- ▶ Une application exécute des conteneurs qui sont immutables (les modifications sont éphémères et ne perdurent pas dans le temps, elles sont en RAM)
- ▶ L'orchestrateur des conteneurs peut à tout moment relancer un POD sur un autre serveur, il y a alors perte de l'état (RAM)



# Anatomie d'une application

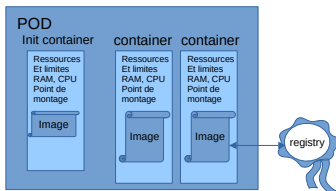
Pour faire perdurer des données hors du conteneur :

- ▶ les Secrets ou les ConfigMaps permettent de créer des données résistantes dans le temps (en lecture seule)
- ▶ Des variables d'environnement peuvent être fixées au niveau du *Deployment*
- ▶ Un système de stockage externe (PVC → Volumes → Point de montage) peut être proposé, en mode
  - ▶ exclusif (bloc : ReadWriteOnce)
  - ▶ concurrent (FS: ReadWriteMany)

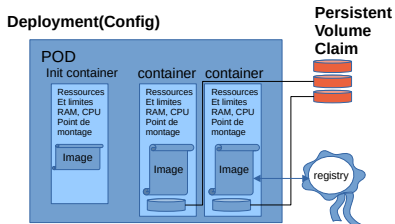


# Anatomie d'une application

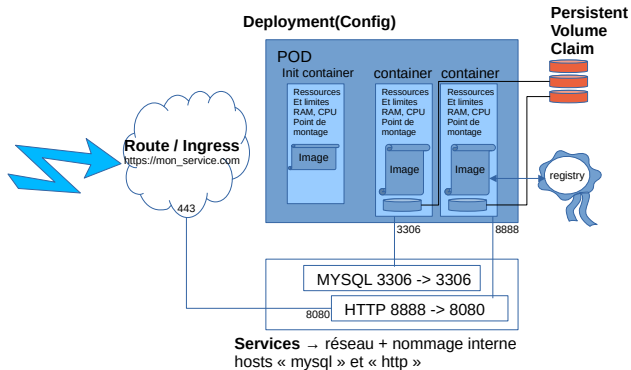
## Deployment(Config)



# Anatomie d'une application



# Anatomie d'une application



# Anatomie d'une application

Il est bon d'avoir en tête les choses suivantes :

- ▶ Un conteneur ne peut pas tourner en tant que **ROOT**
- ▶ L'utilisateur courant est arbitraire et est du groupe **ROOT**
- ▶ Par convention, le fichier Dockerfile se termine par **USER 1001**
- ▶ Bonne façon de gérer les permissions en écriture pour l'utilisateur arbitraire : **chgrp -R root dossier && chown -R g=u dossier**
- ▶ Les ports réseaux des services sont publics ( $> 1024$  )

# Anatomie d'une application

PLMshift met à disposition un ensemble d'images prêtes à l'emploi et bien configurées : **oc get is -n openshift**

- ▶ nginx
- ▶ apache
- ▶ php
- ▶ nodejs
- ▶ ruby
- ▶ python
- ▶ mysql, mariadb, postgresql,
- ▶ mongodb, redis
- ▶ ...



# La sauvegarde des applications

- ▶ Supprimer une application est aussi simple que déployer une application...
- ▶ Vous pouvez donc à tout moment détruire l'ensemble des données de vos applications
- ▶ Vous devez donc vous assurer d'avoir une sauvegarde de votre application
- ▶ Ne vous appuyez pas uniquement sur les mécanismes de sauvegarde de la PLM qui peuvent être défectueux



# La sauvegarde des applications

- ▶ voir l'exemple de **webdav** sur `https://plmlab.math.cnrs.fr/plmshift/wordpress`
- ▶ associé à des **CronJobs** comme sur `https://plmshift.pages.math.cnrs.fr/outils_de_consolidation/mysqldump/`

# Développer des applications

Développer une application revient à :

- ▶ Fabriquer un conteneur en éditant un Dockerfile et en stockant l'image dans un registry externe
- ▶ Compléter un conteneur existant avec un dépôt GIT et des scripts (S2I) et en stockant l'image dans le registry interne

Lire impérativement la documentation <https://docs.okd.io/latest/cicd/builds/understanding-buildconfigs.html>

## Développer des applications = personnaliser des images (format docker) pour un déploiement

Principe : à partir d'une image source, obtenir une image personnalisée prête à être déployée. Pour cela OKD propose une facilité pour le développeur, les **BuildConfigs**

- ▶ Composition de l'image personnalisée
  - ▶ un **BuildConfig** avec une recette
    - ▶ **strategy Docker** : dockerfile
    - ▶ **strategy Source** : s2i (source2image)
  - ▶ un **Image Stream** = stockage de l'image de référence dans le projet utilisateur (registry local au projet)

Lire impérativement la documentation

[https://docs.okd.io/latest/dev\\_guide/builds/index.html](https://docs.okd.io/latest/dev_guide/builds/index.html)

# Éléments d'un BuildConfig

## Principe :

On part d'une image Docker quelconque (quay, docker.io, etc.) ou bien d'une image stockée dans le registry interne :  
**from Dockerfile ou from ImageStreamTag**

On décrit les modifications dans une recette Docker ou bien on fusionne avec le contenu d'un dépôt GIT tiers :  
**Strategy Type Docker ou Strategy Type Source**

L'image résultante est déposée dans le registry réservé au projet :  
**to : ImageStreamTag**



# Exemple de BuildConfig basé sur une image Docker

```
kind: BuildConfig
apiVersion: v1
metadata:
  name: application
  labels:
    app: application
spec:
  source:
    type: Git
    git:
      uri: https://plmlab.math.cnrs.fr/plmshift/hugo.git
      ref: master
      contextDir: hug-s2i
  strategy:
    type: Docker
    dockerStrategy:
      from:
        kind: DockerImage
        name: golang:1.13-buster
  output:
    to:
      kind: ImageStreamTag
      name: application:latest
```

```
kind: ImageStream
apiVersion: v1
metadata:
  name: application
  labels:
    app: application
```



# Exemple de BuildConfig basé sur Source2Image

```
kind: BuildConfig
apiVersion: v1
metadata:
  name: monlamp
  labels:
    app: monapp
spec:
  triggers:
  - type: ConfigChange
  - type: ImageChange
  source:
    type: Git
    git:
      uri: https://plmlab.math.cnrs.fr/plmshift/drupal.git
      ref: master
  strategy:
    type: Source
    sourceStrategy:
      from:
        kind: ImageStreamTag
        namespace: openshift
        name: php:7.3
  output:
    to:
      kind: ImageStreamTag
      name: monlamp:latest
```

```
kind: ImageStream
apiVersion: v1
metadata:
  name: monlamp
  labels:
    app: monapp
```



# Rien ne fonctionne comme prévu

PLMshift est une plateforme expérimentale, basée sur OKD, la version libre d'OpenShift de RedHat. Elle évolue rapidement au cours du temps, il est possible que la documentation soit un temps en retard.

PLMshift est un outil complexe, puissant, utilisant des concepts nouveaux. Les utilisateurs peuvent s'appuyer sur une documentation foisonnante :

- ▶ <https://plmshift.pages.math.cnrs.fr>
- ▶ <https://docs.okd.io/latest>
- ▶ <https://cookbook.openshift.org/>
- ▶ <https://cloud.redhat.com/blog/tag/how-tos/>

De plus, à travers l'interface Web et la ligne de commande il est possible d'obtenir un très grand nombre d'informations.

Il est donc tout à fait possible de s'y perdre... (euphémisme)



# Rien ne fonctionne comme prévu

- ▶ Chaque projet a un quota de ressources pré alloué. Ce quota est volontairement réduit, donc rapidement vos processus ne se lanceront pas... → sollicitez Support pour augmenter ces quotas
- ▶ Docker.io a récemment mis en place une limitation de chargement d'images de conteneurs depuis leur catalogue. Voici les préconisations pour contourner cette limite sur le forum Mathrice
- ▶ Prenez le temps de lire les documentations et blogs et surtout rencontrez nous sur <https://mattermost.math.cnrs.fr/plm-support/channels/plmshift>
- ▶ Et enfin, évitez de faire fonctionner votre application de manière *historique* (comme sur une VM...comme faire tourner un cron dans le Pod, l'objet CronJob est là pour ça)





# Rien ne fonctionne comme prévu

- ▶ PLMshift redirige tout le trafic HTTP en HTTPS, il faut donc veiller à ce que les routes aient le TLS activé :

```
kind: Route
apiVersion: route.openshift.io/v1
...
spec:
  tls:
    termination: edge
```

- ▶ Pour des noms de domaine différents de *quelquechose.apps.math.cnrs.fr*, contactez Support pour que PLMshift accepte ce nouveau domaine.
- ▶ PLMshift peut générer automatiquement des certificats HTTPS *LetsEcnrypt* ([https://plmshift.pages.math.cnrs.fr/Foire\\_aux\\_Questions/certificat/](https://plmshift.pages.math.cnrs.fr/Foire_aux_Questions/certificat/))



# Le futur d'OKD dans la PLM

- ▶ Plusieurs instances d'OKD, pour des applications de type différent (NoteBook, expérimentation, développement, production)
- ▶ Avec des niveaux de services adaptés
- ▶ Déployés sous forme d'application sur OpenStack
- ▶ Permet à tout un chacun de participer sans privilèges particuliers (utilisateur, contributeur, administrateur, ...)



# Offre de services Notebooks / jupyterHub

L'offre est visible ici <https://jupytercloud.math.cnrs.fr>

jupytercloud.math.cnrs.fr

## JupyterCloud

Jupyter as a Service

DÉCOUVRIR LES PLATEFORMES

DÉPLOYER MA PROPRE INSTANCE



### Accéder

Utiliser un ordinateur ou une tablette pour lancer des calculs depuis un navigateur WEB



### S'authentifier

Utiliser son compte d'établissement pour s'authentifier sur la Fédération d'Instituts Éducation-Recherche opérée par le Groupement d'Intérêt Public ICNATEL



### Paramétrer

Choisir une image docker métier enregistrée dans le registre docker de JupyterCloud



### Calculer

Exécuter ses notebooks Jupyter au sein d'un centre de données et de calcul français



### Collaborer

Partager ses notebooks Jupyter sur l'annuaire de notebook de JupyterCloud



### Accéder aux données

Connecter des partages sur n'importe quelle infrastructure de stockage compatible avec les protocoles

[WebDAV](#) [Deps S3](#)

### Le projet

Jupyter as a Service  
Réseau Thématique MATHRISE

### Partenaires

Groupe CALCUL  
SERVICE F3-CLOUD  
MÉSOCENTRE GRID4  
MÉSOCENTRE VirtualData

### Technologies

Shibboleth / OpenID Connect  
JupyterHub / Jupyter  
Docker / Podman  
Kubernetes / Helm  
DVC / OperatorHub  
OpenStack



## JupyterCloud

La PLM propose différentes plateformes pour des publics variés.



### Cloud VirtualData

JupyterCloud

SE CONNECTER

- 🖨 96 CPUs
- 📄 192 Go RAM
- 📁 1 Go de RAM et 2 Go de Disque par notebook

🔒 Une instance JupyterHub réservée à la communauté mathématiques française



### Cloud PLMshift

JupyterHub

SE CONNECTER

- 🖨 200 CPUs
- 📄 512 Go RAM
- 📁 2 Go de RAM et 2 Go de Disque par notebook
- 🔌 10 vGPUs

🔒 Une instance JupyterHub réservée à la communauté mathématiques française

PimDataHub

SE CONNECTER

- 🖨 100 CPUs
- 📄 256 Go RAM
- 📁 2 Go de RAM et 2 Go de Disque par notebook
- 🔌 10 vGPUs

🌐 Une instance JupyterHub ouverte à la communauté enseignement supérieur recherche

Consultez [la FAQ](#) pour mieux utiliser ces instances.

Vous avez un compte PLM ? Alors retrouvez-nous sur [Mattermost](#)

# Offre de services Notebooks

🔒 jupytercloud.math.cnrs.fr/instances/



FAQ Accueil Les plateformes Ma propre instance

## JupyterCloud

La PLM vous permet d'instancier votre propre JupyterHub ou mettre vos Notebooks en ligne



### Mon instance Jupyter Hub

DÉCOUVRIR

- 🔒 Service réservé à la communauté mathématiques française
- 🔑 Authentification dédiée aux comptes de la PLM
- 🔑 Authentification ouverte à l'ensemble de la Fédération RENATER
- 🔑 Un grand nombre de noyaux disponibles
- 🔑 Accès à des fractions de GPUs



### Mes Noyaux customisés

DÉCOUVRIR

- 🔒 Service réservé à la communauté mathématiques française
- ↔ Configurable via un dépôt GIT
- 🔑 Basé sur [la bibliothèque Thamos](#)
- 🔑 Permet d'étendre simplement les noyaux disponibles dans votre instance Jupyter Hub



### Des Notebooks en ligne

DÉCOUVRIR

- 🔒 Service réservé à la communauté mathématiques française
- 🔑 Partagez vos travaux simplement
- 🔑 Développez vos futurs noyaux Jupyter Hub

Vous avez un compte PLM ? Alors retrouvez-nous sur [Mattemost](#)

