



RÉPUBLIQUE
FRANÇAISE

*Liberté
Égalité
Fraternité*

Inserm

La science pour la santé
From science to health



UTILISATION DE CONDA

AVEC LA PLATEFORME HPC INSERM

Version 0.1

Auteur : Sébastien Aucouturier (ITQ)

Conda	3
Présentation de Conda	3
Utilisation de Conda avec la plateforme HPC	3
Étape 1 : Configuration du serveur Nexus	3
Étape 2 : Utilisation des environnements Conda	3
1.1 Création d'un environnement	4
1.2 Activation d'un environnement	4
1.3 Installer les paquets	5
1.4 Désactivation de l'environnement	6
Étape 3 : Vérification sur un nœud de calcul GPU	7
Étape 4 : Utilisation de l'environnement Conda avec Jupyter	7
Étape 5 : Vérification de l'environnement Conda depuis Jupyter	8
Limitation importante : installation de paquets depuis Jupyter	9
Conda et PIP	9
Conclusion	10

CONDA

PRESENTATION DE CONDA

Conda est un gestionnaire d'environnements et de paquets largement utilisé en **science des données**, **HPC** et **machine learning**.

Contrairement à pip, Conda peut installer :

- des bibliothèques **Python** ;
- des **binaires système** ;
- des bibliothèques **C/C++** utilisées par Python.

Cela en fait un outil particulièrement adapté aux environnements HPC et aux applications nécessitant des dépendances complexes (CUDA, BLAS, MKL, etc.).

UTILISATION DE CONDA AVEC LA PLATEFORME HPC

ÉTAPE 1 : CONFIGURATION DU SERVEUR NEXUS

Comme pour l'installation de paquets Python ou R, la plateforme HPC s'appuie sur un **serveur Nexus** pour le téléchargement des paquets Conda.

Si cette configuration n'a pas encore été effectuée, il est nécessaire de la mettre en place sur votre compte HPC.

Référence : pages 7 à 9 du document [INSERM_CLOUD_TUTO_HPC_VLLM.PDF](#)

ÉTAPE 2 : UTILISATION DES ENVIRONNEMENTS CONDA

Il est fortement recommandé d'utiliser des **environnements Conda** afin d'isoler chaque projet.

Chaque projet peut disposer :

- de sa propre version de Python ;
- de ses propres bibliothèques ;
- sans conflit avec les autres projets.

EXEMPLE

- **Projet A** → Python 3.8 + version ancienne de NumPy
- **Projet B** → Python 3.11 + version récente de NumPy

Aucun conflit entre les projets.

1. CREATION ET GESTION DES ENVIRONNEMENTS

1.1 CREATION D'UN ENVIRONNEMENT

La création d'un environnement Conda se fait via la commande suivante :

```
$conda create --name conda_env_test -y
```

```
[saucouturier@ood ~]$ conda create -n conda-env-test -y
# Please Terms of Service accepted
Channels:
- https://nexus.hds.inserm.fr/repository/conda-pytorch-proxy
- https://nexus.hds.inserm.fr/repository/conda-nvidia-proxy
- https://nexus.hds.inserm.fr/repository/conda-proxy-main
- https://nexus.hds.inserm.fr/repository/conda-forge-proxy
Platform: linux-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: /home/saucouturier/.conda/envs/conda-env-test

Downloading and Extracting Packages:

Preparing transaction: done
Verifying transaction: done
Executing transaction: done

#
# To activate this environment, use
#
#   $ conda activate conda-env-test
#
# To deactivate an active environment, use
#
#   $ conda deactivate
```

conda_env_test correspond au nom de l'environnement.

1.2 ACTIVATION D'UN ENVIRONNEMENT

```
$conda activate conda_env_test
```

Il est possible que cette commande retourne une erreur liée à l'initialisation de Conda.

```
[saucouturier@podsgpu-a100-0 ~]$ conda activate conda-env-test
CondaError: Run 'conda init' before 'conda activate'
[saucouturier@podsgpu-a100-0 ~]$
```

Il est déconseillé d'exécuter : `$conda init`

Cette commande modifie le fichier `.bashrc` et rend le Python de Conda prioritaire sur celui du nœud de calcul, ce qui peut provoquer des comportements indésirables sur le HPC.

SOLUTION RECOMMANDEE

La commande suivante permet d'initialiser Conda sans modifier durablement l'environnement : `$ eval "$(conda shell.bash hook)"`

Afin d'éviter de devoir mémoriser cette commande, il est conseillé de la définir sous forme d'**alias** à la fin du fichier `.bashrc`, par exemple :

```
alias conda-init='eval "$(conda shell.bash hook)''
```

```
# >>> conda initialize >>>
# !! Contents within this block are managed by 'conda init' !!
export PATH="/opt/miniconda3/condabin:$PATH"
# <<< conda initialize <<<
alias conda_init='eval "$(conda shell.bash hook)"'
```

Après l'ajout, rechargez votre profil :

```
$source ~/.bashrc
```

Vous pouvez alors activer Conda proprement :

```
$conda-init
```

```
$conda activate conda_env_test
```

```
[saucoururier@ood ~]$ conda_init
(base) [saucoururier@ood ~]$ conda activate conda-env-test
(conda-env-test) [saucoururier@ood ~]$
```

1.3 INSTALLER LES PAQUETS

Nous pouvons maintenant installer des paquets dans l'environnement Conda actif.

Dans cet exemple, nous installons le paquet **PyTorch** avec **prise en charge des GPU**, en utilisant **Python 3.11** comme version de l'interpréteur :

```
$conda install python=3.11 pytorch-gpu -y
```

Utilisation de Conda avec la plateforme HPC Inserm

```
(conda-env-test) [saucouturier@ood ~]$ conda install python3.11 pytorch-gpu -y
1 channel(s) of service(s) activated
Channels:
 - https://nexus.hds.inserm.fr/repository/conda-pytorch-proxy
 - https://nexus.hds.inserm.fr/repository/conda-nvidia-proxy
 - https://nexus.hds.inserm.fr/repository/conda-proxy-main
 - https://nexus.hds.inserm.fr/repository/conda-forge-proxy
Platform: linux-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

environment location: /home/saucouturier/.conda/envs/conda-env-test

added / updated specs:
 - python3.11
 - pytorch-gpu

The following NEW packages will be INSTALLED:

_libgcc_mutex             repository/conda-proxy-main/linux-64:_libgcc_mutex-0.1-main
_openmp_mutex            repository/conda-proxy-main/linux-64:_openmp_mutex-5.1.1-gnu
binutils_impl_linux64    repository/conda-proxy-main/linux-64:binutils_impl_linux64-2.44-hb9a079_2
blas                     repository/conda-proxy-main/linux-64:blas-1.0-mkl
bz2ip2                   repository/conda-proxy-main/linux-64:bz2ip2-1.0.8-h5ee18b_6
ca-certificates          repository/conda-proxy-main/linux-64:ca-certificates-2023.12.2-h664380_0
cuda-cccl_linux64        repository/conda-nvidia-proxy/linux-64:cuda-cccl_linux64-12.9.27-0
cuda-crt-devel_linux64  repository/conda-nvidia-proxy/nvarch:cuda-crt-devel_linux64-12.9.86-0
cuda-crt-tools           repository/conda-nvidia-proxy/linux-64:cuda-crt-tools-12.9.86-0
cuda-cudart              repository/conda-nvidia-proxy/linux-64:cuda-cudart-12.9.79-0
cuda-cudart-devel_linux64 repository/conda-nvidia-proxy/linux-64:cuda-cudart-devel_linux64-12.9.79-0
cuda-cudart-static       repository/conda-nvidia-proxy/nvarch:cuda-cudart-static_linux64-12.9.79-0
cuda-cudart-static_linux64 repository/conda-nvidia-proxy/nvarch:cuda-cudart-static_linux64-12.9.79-0
cuda-cuobjdump           repository/conda-nvidia-proxy/linux-64:cuda-cuobjdump-12.9.82-1
cuda-cupti               repository/conda-nvidia-proxy/linux-64:cuda-cupti-12.9.79-0
cuda-driver-devel_linux64 repository/conda-nvidia-proxy/nvarch:cuda-driver-devel_linux64-12.9.79-0
cuda-nvcc                repository/conda-nvidia-proxy/nvarch:cuda-nvcc-devel_linux64-12.9.86-0
cuda-nvcc-devel_linux64 repository/conda-nvidia-proxy/nvarch:cuda-nvcc-devel_linux64-12.9.86-0
cuda-nvcc-impl           repository/conda-nvidia-proxy/linux-64:cuda-nvcc-impl-12.9.86-0
cuda-nvcc-tools          repository/conda-nvidia-proxy/linux-64:cuda-nvcc-tools-12.9.86-0
cuda-nvml_c_linux64      repository/conda-nvidia-proxy/linux-64:cuda-nvml_c_linux64-12.9.86-0
cuda-nvmlasm             repository/conda-nvidia-proxy/linux-64:cuda-nvmlasm-12.9.86-1
cuda-nvrtc               repository/conda-nvidia-proxy/linux-64:cuda-nvrtc-12.9.86-0
cuda-nvtx                repository/conda-nvidia-proxy/linux-64:cuda-nvtx-12.9.79-0
cuda-nvml-dev_linux64   repository/conda-nvidia-proxy/nvarch:cuda-nvml-dev_linux64-12.9.86-0
cuda-nvml-impl           repository/conda-nvidia-proxy/linux-64:cuda-nvml-impl-12.9.86-0
cuda-nvml-tools          repository/conda-nvidia-proxy/linux-64:cuda-nvml-tools-12.9.86-0
cuda-version             repository/conda-nvidia-proxy/nvarch:cuda-version-12.9.3
cudnn                    repository/conda-proxy-main/linux-64:cudnn-9.1.7.29-1h76022_0
cudat                    repository/conda-proxy-main/linux-64:cudat-2.4.4-0754ed3_0
filelock                 repository/conda-proxy-main/linux-64:filelock-3.20.3-py311h66a308_0
fsspec                   repository/conda-proxy-main/linux-64:fsspec-2024.10-py311h66a308_0
gcc_impl_linux64         repository/conda-forge-proxy/linux-64:gcc_impl_linux64-13.4.0-h60c5793_7
gcc_linux64              repository/conda-forge-proxy/linux-64:gcc_linux64-13.4.0-h60c5793_7
gmp                      repository/conda-proxy-main/linux-64:gmp-6.3.0-h6b78d5_0
gmpy2                    repository/conda-proxy-main/linux-64:gmpy2-2.2.2-py311h7865c_0
gxx_impl_linux64         repository/conda-forge-proxy/linux-64:gxx_impl_linux64-13.4.0-haf17267_7
gsx_linux64              repository/conda-forge-proxy/linux-64:gsx_linux64-13.4.0-h87879e_21
icu                       repository/conda-proxy-main/linux-64:icu-73.1-h64678d5_0
importlib-metadata       repository/conda-proxy-main/linux-64:importlib-metadata-8.7.0-py311h66a308_0
intel-openmp             repository/conda-proxy-main/linux-64:intel-openmp-2025.4.0-h664380_172
jinj2                    repository/conda-proxy-main/linux-64:jinj2-3.1.6-py311h66a308_0
kernel-headers_linux64  repository/conda-proxy-main/nvarch:kernel-headers_linux64-1.0.0-h3108a97_1
libelf                   repository/conda-proxy-main/linux-64:libelf-1.0.15-h664380_2
libb2seal                repository/conda-proxy-main/linux-64:libb2seal-20250814.1-cxx17-hc842e_0
libcublas               repository/conda-nvidia-proxy/linux-64:libcublas-12.9.1.4-0
libcudam                repository/conda-nvidia-proxy/linux-64:libcudam-12.9.1.4-0
libcudam-devel           repository/conda-nvidia-proxy/linux-64:libcudam-devel-12.9.1.4-0
libcufft                 repository/conda-nvidia-proxy/linux-64:libcufft-11.4.1.4-0
libcurand               repository/conda-nvidia-proxy/linux-64:libcurand-10.3.10-19-0
libcusolver              repository/conda-nvidia-proxy/linux-64:libcusolver-11.7.5.84-0
libcusparse              repository/conda-nvidia-proxy/linux-64:libcusparse-12.5.18.45-0
libcurand               repository/conda-nvidia-proxy/linux-64:libcurand-10.3.10-19-0
libcusolver              repository/conda-nvidia-proxy/linux-64:libcusolver-11.7.5.84-0
libcusparse              repository/conda-nvidia-proxy/linux-64:libcusparse-12.5.18.45-0
libexpat                 repository/conda-proxy-main/linux-64:libexpat-2.7.4-h7354ed3_0
libffi                   repository/conda-proxy-main/linux-64:libffi-3.4.4-h64678d5_1
libgcc                   repository/conda-proxy-main/linux-64:libgcc-15.2.0-h664380_7
libgcc-devel_linux64    repository/conda-forge-proxy/nvarch:libgcc-devel_linux64-13.4.0-h664380_7
libgcc-ng                repository/conda-proxy-main/linux-64:libgcc-ng-15.2.0-h664380_7
libgomp                  repository/conda-proxy-main/linux-64:libgomp-15.2.0-h664380_7
libhwloc                 repository/conda-proxy-main/linux-64:libhwloc-2.12.1-defaul_hfbb79_1000
liblsm                    repository/conda-proxy-main/linux-64:liblsm-2.0.0-h664380_0
libm2t                   repository/conda-nvidia-proxy/linux-64:libm2t-12.9.86-0
libprotobuf              repository/conda-proxy-main/linux-64:libprotobuf-3.31.0-h8732af_1
libramm                  repository/conda-forge-proxy/linux-64:libramm-13.4.0-h664380_7
libstdc++                repository/conda-proxy-main/linux-64:libstdc++-15.2.0-h664380_7
libstdc++-devel_linux64 repository/conda-forge-proxy/nvarch:libstdc++-devel_linux64-13.4.0-h664380_7
libstdc++-ng            repository/conda-proxy-main/linux-64:libstdc++-ng-15.2.0-h664380_7
libtorch                repository/conda-proxy-main/linux-64:libtorch-2.7.0-gpu_cuda128_h3679332_383
libuuid                  repository/conda-proxy-main/linux-64:libuuid-1.41.5-h5ee18b_0
libx11                   repository/conda-proxy-main/linux-64:libx11-1.8.10-h664380_0
libxcb                   repository/conda-proxy-main/linux-64:libxcb-1.17.0-h664380_0
libxml2                  repository/conda-proxy-main/linux-64:libxml2-2.13.9-c2c480e_0
libzlib                  repository/conda-proxy-main/linux-64:libzlib-1.3.1-h664380_0
lit                       repository/conda-proxy-main/linux-64:lit-21.1.1-py311h66a308_0
mapio                    repository/conda-proxy-main/linux-64:mapio-2.0.0-h664380_0
mapio                    repository/conda-proxy-main/linux-64:mapio-2.0.0-h664380_0
mkl                      repository/conda-proxy-main/linux-64:mkl-2025.0.0-h664380_341
mkl-service              repository/conda-proxy-main/linux-64:mkl-service-2.5.2-py311h66a308_0
mkl_fft                  repository/conda-proxy-main/linux-64:mkl_fft-2.1.1-py311h66a308_0
mkl_random               repository/conda-proxy-main/linux-64:mkl_random-1.3.0-py311h66a308_0
mpc                      repository/conda-proxy-main/linux-64:mpc-1.3.1-h664380_0
mpfr                     repository/conda-proxy-main/linux-64:mpfr-4.2.1-h5ee18b_0
mpmath                   repository/conda-proxy-main/linux-64:mpmath-1.3.0-py311h66a308_0
nccl                     repository/conda-nvidia-proxy/linux-64:nccl-2.29.0-h3108a97_1
ncurses                  repository/conda-proxy-main/linux-64:ncurses-6.5-h734776_0
networkx                  repository/conda-proxy-main/linux-64:networkx-3.6.1-py311h66a308_0
numpy                    repository/conda-proxy-main/linux-64:numpy-2.4.2-py311h66a308_0
numpy-base              repository/conda-proxy-main/linux-64:numpy-base-2.4.2-py311h66a308_0
opencl                    repository/conda-proxy-main/linux-64:opencl-3.0.13-h664380_0
opencl-headers           repository/conda-proxy-main/linux-64:opencl-headers-3.0.13-h664380_0
packaging                repository/conda-proxy-main/linux-64:packaging-25.0-py311h66a308_1
pip                      repository/conda-proxy-main/linux-64:pip-26.0.1-py311h66a308_0
pthread-stubs            repository/conda-proxy-main/linux-64:pthread-stubs-0.3-h664380_1
python                   repository/conda-proxy-main/linux-64:python-3.11.14-h664380_0
pytorch                  repository/conda-proxy-main/linux-64:pytorch-2.7.0-gpu_cuda128_py311h66a308_383
pytorch-gpu              repository/conda-proxy-main/linux-64:pytorch-gpu-2.7.0-gpu_cuda128_hf285fa6_383
readline                 repository/conda-proxy-main/linux-64:readline-8.3-hc2a1286_0
setuptools                repository/conda-proxy-main/linux-64:setuptools-75.0.0-py311h66a308_0
sleep                    repository/conda-proxy-main/linux-64:sleep-3.5.1-h5ee18b_2
sqlite                   repository/conda-proxy-main/linux-64:sqlite-3.51.1-h664380_0
swig                      repository/conda-proxy-main/linux-64:swig-4.1.4-py311h66a308_1
sysroot_linux64         repository/conda-proxy-main/nvarch:sysroot_linux64-2.28-h3108a97_1
tblib-devel              repository/conda-proxy-main/linux-64:tblib-devel-2022.2.0-h664380_1
tk                        repository/conda-proxy-main/linux-64:tk-8.6.15-h54e8a7_0
tintin                   repository/conda-proxy-main/linux-64:tintin-3.0.0-h664380_1
typing-extensions         repository/conda-proxy-main/linux-64:typing-extensions-4.15.0-py311h66a308_0
typing_extensions       repository/conda-proxy-main/nvarch:typing_extensions-4.15.0-h664380_0
wheel                    repository/conda-proxy-main/linux-64:wheel-0.46.3-py311h66a308_0
xorg-libx11              repository/conda-proxy-main/linux-64:xorg-libx11-1.8.12-h664380_1
xorg-libxau              repository/conda-proxy-main/linux-64:xorg-libxau-1.8.12-h664380_0
xorg-libxdmcp            repository/conda-proxy-main/linux-64:xorg-libxdmcp-1.1.5-h664380_0
xorg-xorgproto           repository/conda-proxy-main/linux-64:xorg-xorgproto-2024.1-h5ee18b_1
xz                       repository/conda-proxy-main/linux-64:xz-5.4.0-h664380_1
zipp                     repository/conda-proxy-main/linux-64:zipp-3.23.0-py311h66a308_0
zlib                     repository/conda-proxy-main/linux-64:zlib-1.3.1-h625080a_0

Downloading and Extracting Packages:
```

1.4 DESACTIVATION DE L'ENVIRONNEMENT

\$conda deactivate

```
(conda-env-test) [saucouturier@ood ~]$ conda deactivate
(base) [saucouturier@ood ~]$
```

ÉTAPE 3 : VERIFICATION SUR UN NŒUD DE CALCUL GPU

- On Alloue Un Nœud Sur La Partition **A100-10** :

```
$srun -p A100-10 --pty bash
```

- Exécution Du Code Test Sans Environnement Conda → Échec Attendu

```
[saucouturier@podsgpu-a100-0 ~]$ python3 /shared/examples/test_torch_presence_gpu.py
Traceback (most recent call last):
  File "/shared/examples/test_torch_presence_gpu.py", line 1, in <module>
    import torch
ModuleNotFoundError: No module named 'torch'
[saucouturier@podsgpu-a100-0 ~]$
```

- Activation De L'environnement

```
[saucouturier@podsgpu-a100-0 ~]$ conda init
(base) [saucouturier@podsgpu-a100-0 ~]$ conda activate conda-env-test
(conda-env-test) [saucouturier@podsgpu-a100-0 ~]$
```

- Nouvelle exécution du code test → Succès

```
(conda-env-test) [saucouturier@podsgpu-a100-0 ~]$ python3 /shared/examples/test_torch_presence_gpu.py
2.7.0
Number of GPUs: 1
Torch CUDA Version: 12.8
GPU 0: NVIDIA A100 80GB PCIe MIG 1g.10gb
Torch: Total Memory (GB): 10.200547328
(conda-env-test) [saucouturier@podsgpu-a100-0 ~]$
```

L'exécution confirme que PyTorch détecte correctement le GPU depuis l'environnement Conda.

ÉTAPE 4 : UTILISATION DE L'ENVIRONNEMENT CONDA AVEC JUPYTER

Pour utiliser un environnement Conda dans Jupyter, il est nécessaire d'installer un noyau (ipykernel).

- Installation du noyau Jupyter

```
$conda-init
```

```
$conda activate conda_env_test
```

```
$conda install -n conda_env_test ipykernel -y
```

- Enregistrement de l'environnement comme kernel Jupyter

```
python3 -m ipykernel install --user \
```

```
--name conda_env_test \
```

```
--display-name "conda-env-test"
```

```
[conda-env-test] [saucourtier@ood ~]$ conda install ipykernel -y
└─ channel Terms of Service accepted
Channels:
- https://nexus.hds.inserm.fr/repository/conda-pytorch-proxy
- https://nexus.hds.inserm.fr/repository/conda-nvidia-proxy
- https://nexus.hds.inserm.fr/repository/conda-proxy-main
- https://nexus.hds.inserm.fr/repository/conda-forge-proxy
Platform: Linux-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

environment location: /home/saucourtier/.conda/envs/conda-env-test
added / updated specs:
  ipykernel

The following NEW packages will be INSTALLED:

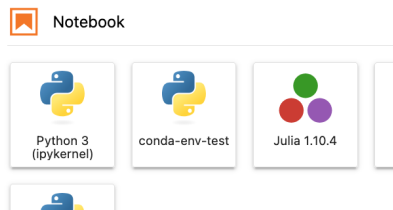
asttokens      repository/conda-proxy-main/linux-64::asttokens-3.0.0-py311h06a4308_0
comm           repository/conda-proxy-main/linux-64::comm-0.2.3-py311h06a4308_0
debugpy        repository/conda-proxy-main/linux-64::debugpy-1.8.16-py311hbdd6827_1
decorator      repository/conda-proxy-main/linux-64::decorator-5.2.1-py311h06a4308_0
executing      repository/conda-proxy-main/linux-64::executing-2.2.1-py311h06a4308_0
ipykernel      repository/conda-proxy-main/linux-64::ipykernel-6.31.0-py311h7040dfc_0
ipython        repository/conda-proxy-main/linux-64::ipython-9.7.0-py311h06a4308_0
ipython_pygmts repository/conda-proxy-main/linux-64::ipython_pygmts_kernr-1.1.1-py311h06a4308_0
jedi           repository/conda-proxy-main/linux-64::jedi-0.19.2-py311h06a4308_0
jupyter_client repository/conda-proxy-main/linux-64::jupyter_client-8.8.0-py311h06a4308_0
jupyter_core   repository/conda-proxy-main/linux-64::jupyter_core-5.9.1-py311h06a4308_0
libiodium      repository/conda-proxy-main/linux-64::libiodium-1.0.20-hex8642_0
matplotlib-inline repository/conda-proxy-main/linux-64::matplotlib-inline-0.2.1-py311h06a4308_0
nest-asyncio   repository/conda-proxy-main/linux-64::nest-asyncio-1.6.0-py311h06a4308_0
parsio         repository/conda-proxy-main/linux-64::parsio-0.5-py311h06a4308_0
pexpect        repository/conda-proxy-main/linux-64::pexpect-4.9.0-py311h06a4308_1
platformdirs  repository/conda-proxy-main/linux-64::platformdirs-4.5.0-py311h06a4308_0
prompt_toolkit repository/conda-proxy-main/linux-64::prompt_toolkit-3.0.22-py311h06a4308_1
prompt_toolkit repository/conda-proxy-main/noarch::prompt_toolkit-3.0.52-hd3eb1b0_1
psutil         repository/conda-proxy-main/linux-64::psutil-7.0.0-py311heeb239_1
ptyprocess     repository/conda-proxy-main/linux-64::ptyprocess-0.7.0-pyhd3eb1b0_3
pure_eval      repository/conda-proxy-main/linux-64::pure_eval-0.2.3-py311h06a4308_0
pygments       repository/conda-proxy-main/linux-64::pygments-2.19.2-py311h06a4308_0
python-dateutil repository/conda-proxy-main/linux-64::python-dateutil-2.9.0post1-py311h06a4308_2
pyzmq          repository/conda-proxy-main/linux-64::pyzmq-27.1.0-py311hcf288c_1
six            repository/conda-proxy-main/linux-64::six-1.17.0-py311h06a4308_0
stock_data     repository/conda-proxy-main/linux-64::stock_data-0.5.3-py311h06a4308_0
tornado        repository/conda-proxy-main/linux-64::tornado-6.5.4-py311h4762149_0
traitlets      repository/conda-proxy-main/linux-64::traitlets-5.14.3-py311h06a4308_0
wcwidth        repository/conda-proxy-main/linux-64::wcwidth-0.2.14-py311h06a4308_0
zeromq         repository/conda-proxy-main/linux-64::zeromq-4.3.5-hb0a5e54_1

Downloading and Extracting Packages:
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
[conda-env-test] [saucourtier@ood ~]$ python3 -m ipykernel install --user --name=conda-env-test --display-name "conda-env-test"
Installed kernelspec conda-env-test in /nfs/inserm/home_users/saucourtier/.local/share/jupyter/kernels/conda-env-test
[conda-env-test] [saucourtier@ood ~]$
```

ÉTAPE 5 : VERIFICATION DE L'ENVIRONNEMENT CONDA DEPUIS JUPYTER

Nous allouons une instance **Jupyter** sur la partition **A100-10** du HPC.

Une fois l'instance démarrée, l'environnement conda-env-test apparaît automatiquement dans la liste des noyaux disponibles.



1. UTILISATION VIA UN NOTEBOOK JUPYTER :

1. Sélectionner le noyau conda-env-test
2. Ouvrir le notebook suivant : /shared/examples/test_torch_presence_gpu.ipynb
3. Exécuter le notebook cellule par cellule

```

[1]: import torch
[2]: print(torch.__version__)
2.7.0
[3]: if torch.cuda.is_available():
    num_gpus = torch.cuda.device_count()
    print(f"Number of GPUs: {num_gpus}")
    print(f"Torch CUDA Version: {torch.version.cuda}")
    for i in range(num_gpus):
        print(f"GPU {i}: {torch.cuda.get_device_name(i)}")
        print(f"Torch: Total Memory (GB): {torch.cuda.get_device_properties(i).total_memory / 1e9}")
    else:
        print("Torch: GPU is not available")
Number of GPUs: 1
Torch CUDA Version: 12.8
GPU 0: NVIDIA A100 80GB PCIe MIG 1g.10gb
Torch: Total Memory (GB): 10.200547328

```

L'exécution confirme que **PyTorch détecte correctement le GPU** depuis l'environnement Conda.

2. UTILISATION DU TERMINAL JUPYTER

Il est également possible d'utiliser le **terminal intégré à Jupyter**.

```

(base) Apptainer> conda_init
(base) Apptainer> conda activate conda-env-test
(conda-env-test) Apptainer> python3 /shared/examples/test_torch_presence_gpu.py
2.7.0
Number of GPUs: 1
Torch CUDA Version: 12.8
GPU 0: NVIDIA A100 80GB PCIe MIG 1g.10gb
Torch: Total Memory (GB): 10.200547328
(conda-env-test) Apptainer>

```

LIMITATION IMPORTANTE : INSTALLATION DE PAQUETS DEPUIS JUPYTER

Depuis une instance Jupyter, l'installation de paquets via **conda**, **pip** ou **R** **n'est pas fonctionnelle**.

Une erreur de type `:NoSpaceLeftError: No space left on device` peut apparaître.

Recommandation

L'installation des paquets doit impérativement être réalisée **en amont**, via le **HPC Shell Access**, avant le lancement de l'instance Jupyter.

CONDA ET PIP

Bien qu'il soit possible d'utiliser **pip** pour installer des paquets dans un environnement **Conda**, cette pratique **n'est pas recommandée**. En effet, pip peut modifier des bibliothèques déjà gérées par Conda, ce qui risque de casser la chaîne de dépendances et de rendre certains paquets **inopérionnels**.

Pour limiter les risques :

1. **Préférer Conda** : Installez d'abord les paquets via Conda lorsque cela est possible.
2. **Utiliser pip avec précaution** : Réservez pip aux paquets introuvables sur les canaux Conda.
3. **Créer des environnements dédiés** : Si vous devez utiliser pip, réalisez-le dans un environnement Conda isolé pour éviter d'impacter d'autres projets.
4. **Installer dans le bon ordre** : Installez d'abord les paquets Conda, puis les paquets pip restants.
5. **Vérifier les dépendances** : Après installation, utilisez `conda list` pour vous assurer qu'aucune librairie critique n'a été remplacée.

CONCLUSION

L'utilisation de Conda sur la plateforme HPC permet :

- une isolation propre des projets ;
- une gestion fiable des dépendances complexes ;
- une compatibilité optimale avec les GPU et les outils de calcul intensif.

Cette approche est fortement recommandée pour les workflows HPC, machine learning et deep learning.