

TURPAN FOR AI



calmip



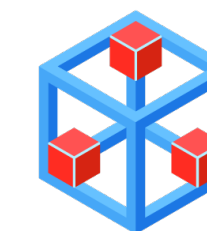
INDEX

Turpan AI environment

Jupyter notebook

Production Mode: Containers

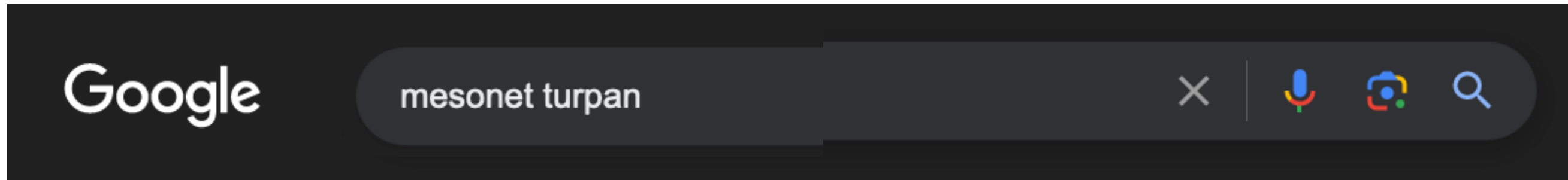
Production Mode: Conda



calmip



TURPAN DOCUMENTATION



<https://www.mesonet.fr/documentation/user-documentation/acces/portail>



Documentation

Portail

Site web

Accueil

Le portail MesoNET

Accéder au portail MesoNET

Gérer ses clés SSH

Créer et gérer des projets

Troubleshooting et FAQ

Pour les formations

Machine Code/Formation

Juliet, machine IA/GPU

Zen

Architectures de Prototypage
et Spécialisées

Boreale, la machine vectorielle

Turpan, la machine ARM

Tutoriaux

Le portail MesoNET

Accéder au portail MesoNET

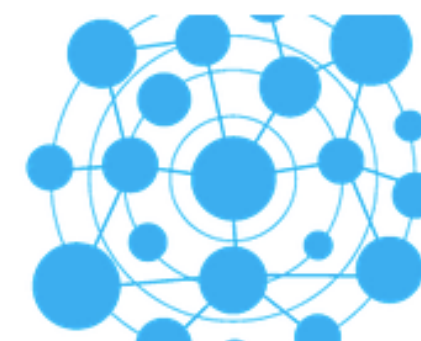
Comment accéder au portail MesoNET ?

Pour accéder au portail MesoNET, il faut au préalable se créer un compte.

Demander un compte MesoNET

Pour demander un compte MesoNET, faites pointer votre navigateur sur l'URL: <https://iam.mesonet.fr/>

1. Si vous le pouvez, **connectez-vous avec vos identifiants institutionnels** (CNRS, université, institutions de recherche étrangères ,etc.) en cliquant sur le **bouton EduGAIN**. Si vous n'avez pas d'identifiants institutionnels (entreprises, ...) et seulement dans ce cas, ouvrez un compte local. Pour cela, visitez l'URL <https://iam.mesonet.fr/start-registration> pour vous créer un compte.
2. Dans tous les cas, vous serez redirigés vers l'écran suivant :




Register at **MesoNET SSO**

This is the MesoNET SSO registration page






TURPAN WEBSITE

 Documentation Portail Site web

Accueil

Le portail MesoNET 


Accéder au portail MesoNET


Gérer ses clés SSH


Créer et gérer des projets


Troubleshooting et FAQ


Pour les formations


Machine Code/Formation 

Juliet, machine IA/GPU 


Zen 

Architectures de Prototypage et Spécialisées 

Boreale, la machine vectorielle 


Turpan, la machine ARM 

Description

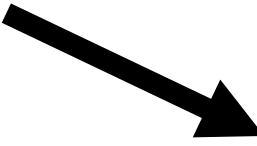

Se connecter à Turpan 

Le stockage


Lancer un calcul

L'accounting 

Environnement de développement

 Les logiciels installés 

Les jeux de données

Améliorer les performances 

Présentation du cluster de prototypage ARM accélérée :Turpan

La machine Turpan est un cluster de prototypage pour le calcul HPC basé sur une architecture ARM Accélérée par des GPUS. Ce cluster est une des machines à architectures expérimentales du projet MesoNET.

[Plus de détails de détails sur le cluster de prototypage Turpan](#)

Faire appel au support

Si quelque chose ne fonctionne pas et que vous n'arrivez pas à vous en sortir, vous pouvez envoyer un courriel à [cette adresse](#), en précisant :

- Le cluster de calcul concerné
- Chemin du répertoire d'appel
- Chemin du script d'appel
- Numéro du job concerné

Remercier le mésocentre Calmip et le projet MesoNET

Vous trouverez ci-dessous deux exemples de remerciement du mésocentre de calcul CALMIP et du projet MesoNET à intégrer dans vos publications :

- version longue : "This work was supported by a French government grant managed by the Agence Nationale de la Recherche under the "Investissements d'avenir" program (reference "ANR-21-ESRE-0051"). This work was granted access to the MesoNET resources center and the MesoNET Project under the allocation m[project_number]."
- version courte : "This work was supported by a French government grant managed by the Agence Nationale de la Recherche under the "Investissements d'avenir" program (reference "ANR-21-ESRE-0051")."





TURPAN WEBSITE



Documentation

Portail

Site web

Architectures de Prototypage et Spécialisées

Boreale, la machine vectorielle

Turpan, la machine ARM

Description

Se connecter à Turpan

Le stockage

Lancer un calcul

L'accounting

Environnement de développement

Les logiciels installés

BLAS

LAPACK

ScaLAPACK

HDF5

PETSc

AmgX

FFTW

NetCDF

Parallel netCDF

Paraview

L'écosystème Python

Conda

Python

Python tools

Architectures de Prototypage et Spécialisées > Turpan, la machine ARM > Les logiciels installés > L'écosystème Python

L'écosystème Python

CALMIP préconise l'utilisation de conda pour l'installation de vos environnements python :

- [La documentation pour conda](#)

Environnements conda prêts à être utilisés ou clonés :

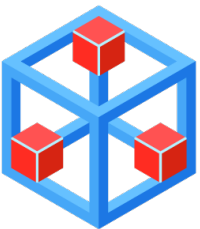
- [python](#) (uniquement python)
- [python-tools](#) (contenant les modules : Numpy, Scipy, Pandas, Matplotlib, JupyterLab plus quelques autres)
- [tensorflow](#) via les conteneurs nvidia
- [pytorch](#) via les conteneurs nvidia
- [scikit-learn](#) et rapids (cuML, cuDF, cuGraph, etc ...) via les conteneurs nvidia
- [modulus](#) via les conteneurs nvidia

Outils interactifs et de visualisation également disponibles :

- [JupyterLab](#)
- [TensorBoard](#)

Previous
« [Paraview](#)

Next
[Conda](#) »



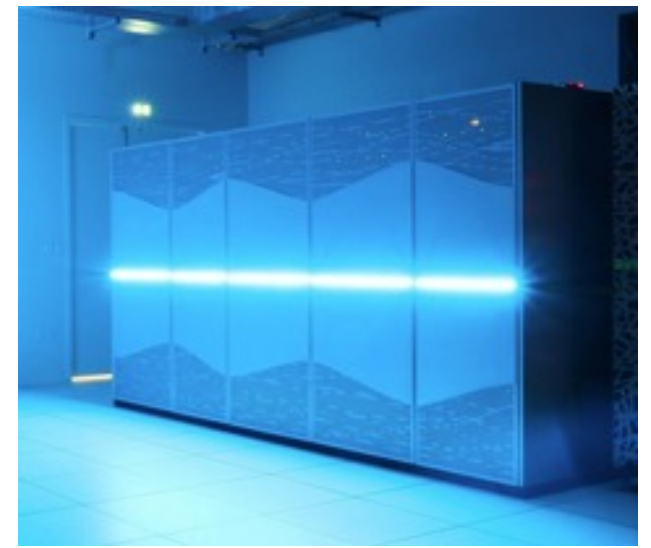
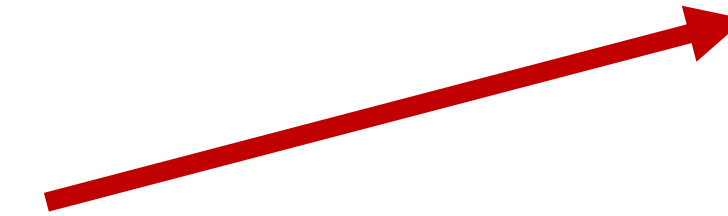
calmip



TURPAN JUPYTER NOTEBOOK

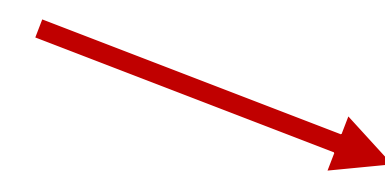
```
[estana@turpanlogin1 estana]$ runJupyterSession.sh
```

```
[estana@turpanlogin1 estana]$ runJupyterSession.sh --container pytorch
```



The screenshot shows the JupyterLab interface. The top bar indicates the current session is 'test.ipynb - JupyterLab'. The left sidebar shows a file browser with a search bar and a list of files: 'libraries', 'softwares', 'tests', and 'test.ipynb'. The main area displays the code editor for 'test.ipynb', showing the following code:

```
[4]: 3 + 8
[4]: 11
[5]: import numpy
[6]: numpy
[6]: <module 'numpy' from '/usr/local/miniconda/22.11.1/envs/python-tools-3.10.9/lib/python3.10/site-packages/numpy/__init__.py'>
[ ]:
```



Allocated resources: 1 A100-GPU, 40 CPUs and 256G RAM





PRODUCTION MODE WITH CONTAINERS (APPTAINER)

Pytorch container example

Container setup

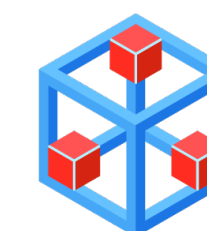
```
apptainer shell --nv /work/conteneurs/sessions-interactive/pytorch-24.02-py3-calmip-si.sif
```

```
Apptainer> pip install --user wandb
```

Running the code using the container

```
#!/bin/bash  
#SBATCH -J mon_job  
#SBATCH -p shared  
#SBATCH --nodes 1  
#SBATCH --ntasks 1  
#SBATCH --time=0:15:00  
#SBATCH --gres=gpu:1
```

```
apptainer exec --bind /tmpdir,/work --nv /work/conteneurs/sessions-interactive/pytorch-24.02-py3-calmip-si.sif python mon_script.py
```



calmip



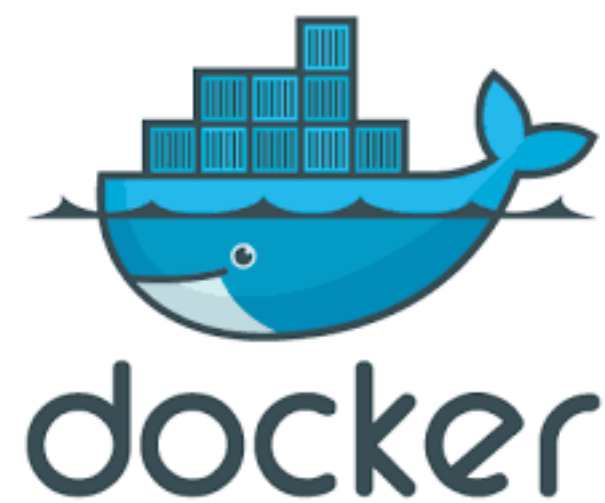
CONTAINERS

A container is essentially a **fully packaged and portable computing environment**

Everything an application required to run :

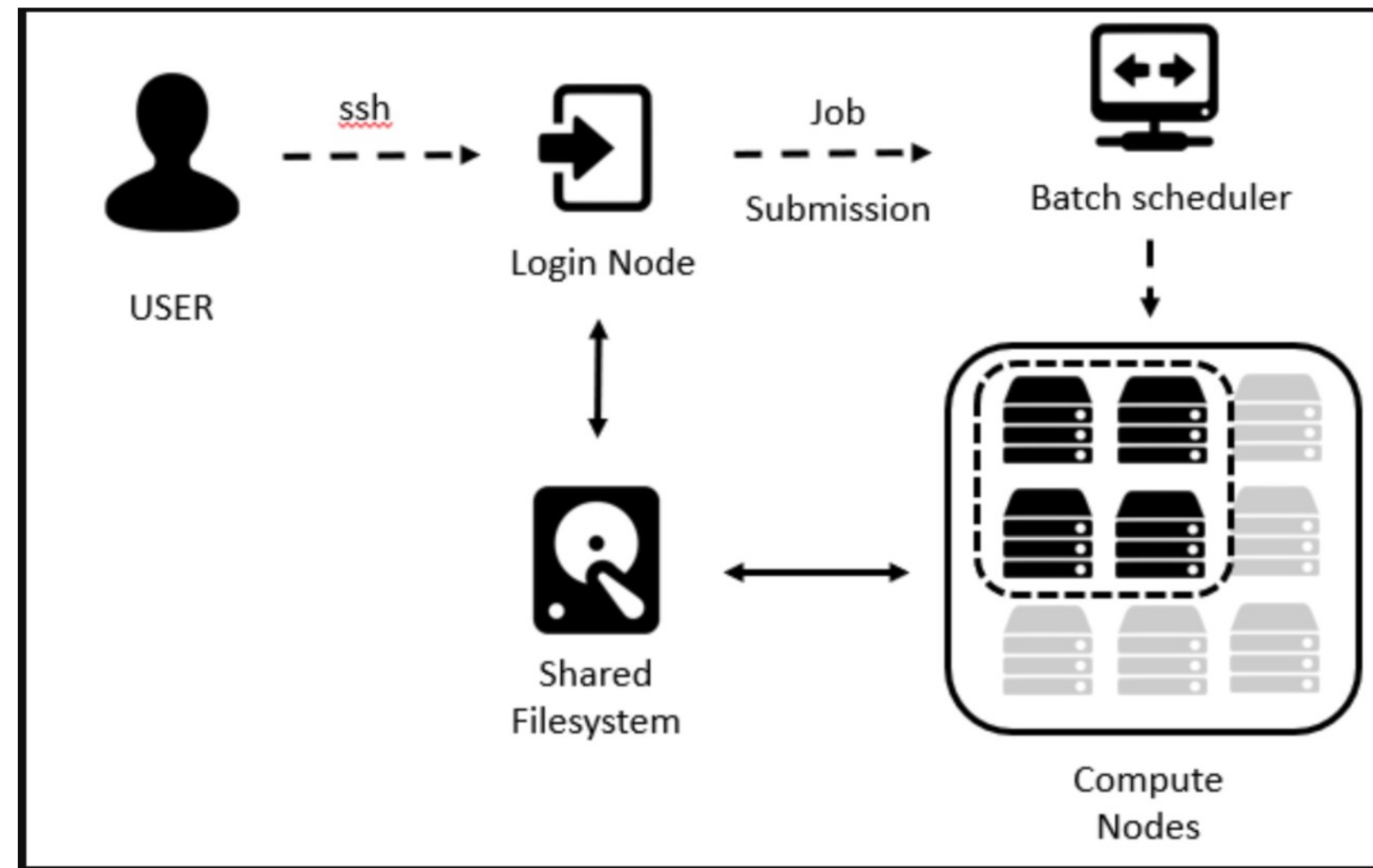
- Binaries
- Libraries
- Configuration files
- Dependencies

Is encapsulated and isolation in its container



BATCH SCHEDULER: SLURM

Batch system



Sbatch script example

```
#!/bin/bash
#SBATCH --job-name=test_mpi
#SBATCH --partition=compute
#SBATCH --ntasks=96
#SBATCH --time=01:30:00
#SBATCH --output=job_%x_%j.out
#SBATCH --err=job_%x_%j.err
#SBATCH --exclusive

# setup environment
module load applications

#launch executable with or without inputs
srun ./mybinary input1 input2
```

Specify the name of the partition

Specify a reasonable time limit

Specify the names of your output files



PRODUCTION MODE WITH CONTAINERS (APPTAINER)

Pytorch container example

Container setup

```
apptainer shell --nv /work/conteneurs/sessions-interactive/pytorch-24.02-py3-calmip-si.sif
```

```
Apptainer> pip install --user wandb
```

Running the code using the container

```
#!/bin/bash
#SBATCH -J mon_job
#SBATCH -p shared
#SBATCH --nodes 1
#SBATCH --ntasks 1
#SBATCH --time=0:15:00
#SBATCH --gres=gpu:1

apptainer exec --bind /tmpdir,/work --nv /work/conteneurs/sessions-interactive/pytorch-24.02-py3-calmip-si.sif python mon_script.py
```





PRODUCTION MODE WITH CONDA

Charger l'environnement

```
module purge  
module load conda/22.11.1
```

Vérifier quels sont les environnements disponibles

```
conda env list
```

```
pytorch-1.10      /usr/local/miniconda/4.9.2/envs/pytorch-1.10  
pytorch-1.11      /usr/local/miniconda/4.9.2/envs/pytorch-1.11  
pytorch-1.12      /usr/local/miniconda/4.9.2/envs/pytorch-1.12  
pytorch-1.7       /usr/local/miniconda/4.9.2/envs/pytorch-1.7  
pytorch-1.7.1     /usr/local/miniconda/4.9.2/envs/pytorch-1.7.1  
pytorch-1.8.0     /usr/local/miniconda/4.9.2/envs/pytorch-1.8.0  
pytorch-1.9.1     /usr/local/miniconda/4.9.2/envs/pytorch-1.9.1  
pytorch-2.0.1     /usr/local/miniconda/4.9.2/envs/pytorch-2.0.1  
rdkit-23.03.1     /usr/local/miniconda/4.9.2/envs/rdkit-23.03.1  
scikit-learn-0.23.2 /usr/local/miniconda/4.9.2/envs/scikit-learn-0.23.2  
tensorflow-1.15   /usr/local/miniconda/4.9.2/envs/tensorflow-1.15  
tensorflow-2.2.0  /usr/local/miniconda/4.9.2/envs/tensorflow-2.2.0  
tensorflow-2.4.1  /usr/local/miniconda/4.9.2/envs/tensorflow-2.4.1
```

Installer des paquets dans l'environnement courant

Pour installer une version spécifique de numpy :

```
(python-tools-3.10.9) [toto@turpanlogin1 ~]$ conda install numpy=1.22
```





QUESTIONS ?

MERCI

