

Getting started with clusters

Jean Feydy
HeKA team, Inria Paris
Inserm, Université Paris-Cité

19th of January, 2023
Team building seminar
Rouen

Why use a cluster?

- **Fast** run times.
- **Large** scale data.
- **Reproducible** experiments.

Feel free to ask me your questions:

- I use clusters **weekly**.
- INRIA Paris: delegate for HeKA at the users' committee.
- INRIA clusters: delegate for Paris at the users' committee.
- Happy to **bring you all** fast run times.

1. Connect to a distant machine with SSH

- SSH authentication
- File transfer with `ssh://` or `sftp://`
- Full remote development with VSCode

2. Clusters that I use regularly – no data security: open or virtual cohorts only!

- Amazon Web Service EC2
- Cleps at INRIA Paris
- Jean Zay, the French cluster for AI research

3. Set up your environment

- pip? conda? Docker?
- Singularity

SSH connection

How to use SSH

Standard tool for remote access:

- Linux and MacOS: ssh is already installed.
- Windows: OpenSSH is officially supported by Microsoft, but opt-in.

Typical commands to open a remote terminal:

- `ssh user@serverip` with password.
- `ssh myfavoritecluster` without password.

I use tmux to create “**immortal**” **shells** on the cluster that stay alive even if I close my session or experience connection issues.

SSH config: ~/.ssh/config or %userprofile%\ssh\config

```
1  # Dummy local connection:
2  Host localhost
3  HostName 127.0.0.1
4  User jean
5
6  # Cleps - Inria Paris:
7  Host cleps
8  User jfeydy
9  HostName cleps.inria.fr
10 port 22
11 ControlMaster auto
12 ForwardX11 no
13 ForwardAgent yes
```

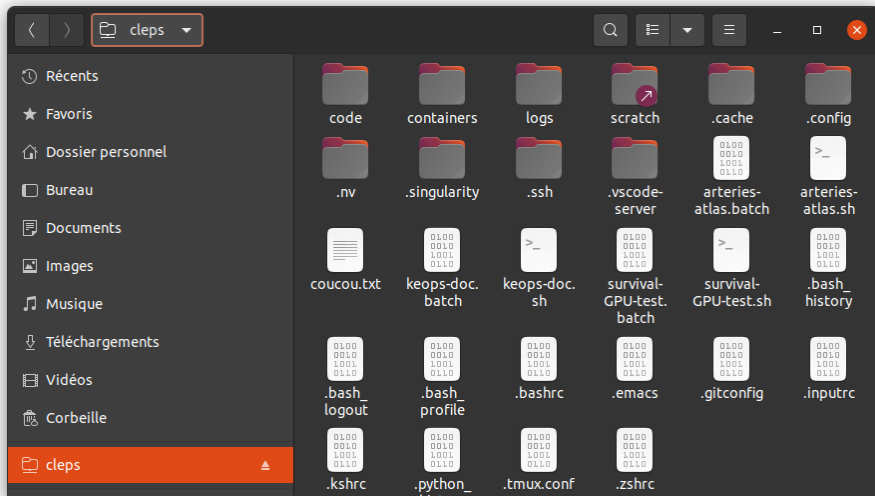
SSH config: ~/.ssh/config or %userprofile%\ssh\config

```
14  # Inria bastion for Jean Zay:
15  Host inria-ssh
16  User jfeydy
17  HostName ssh.paris.inria.fr
18
19  # Jean Zay:
20  Host jean-zay
21  User my_personal_id
22  HostName jean-zay.idris.fr
23  ProxyJump inria-ssh
24
25
26
```

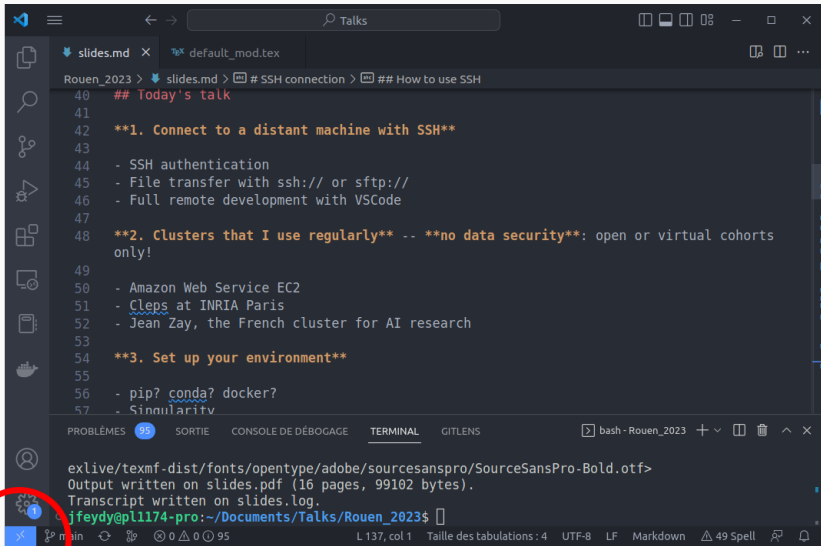
Avoid passwords with a secure key: `~/.ssh/id_rsa` and `id_rsa.pub`

```
1  # Create your id_rsa, if you don't have one already:
2  ssh-keygen -t rsa
3
4  # Ensure that your favorite servers remember you:
5  ssh-copy-id id@server
```


ssh://cleps/home/jfeydy or sftp://... in the address bar



Full remote integration with Visual Studio Code



The screenshot shows the Visual Studio Code interface with a remote SSH connection. The editor displays a markdown file named 'slides.md' with the following content:

```
Rouen_2023 > slides.md > # SSH connection > ## How to use SSH
40 ## Today's talk
41
42 **1. Connect to a distant machine with SSH**
43
44 - SSH authentication
45 - File transfer with ssh:// or sftp://
46 - Full remote development with VSCode
47
48 **2. Clusters that I use regularly** -- **no data security**: open or virtual cohorts
   only!
49
50 - Amazon Web Service EC2
51 - Cleps at INRIA Paris
52 - Jean Zay, the French cluster for AI research
53
54 **3. Set up your environment**
55
56 - pip? conda? docker?
57 - Singularity
```

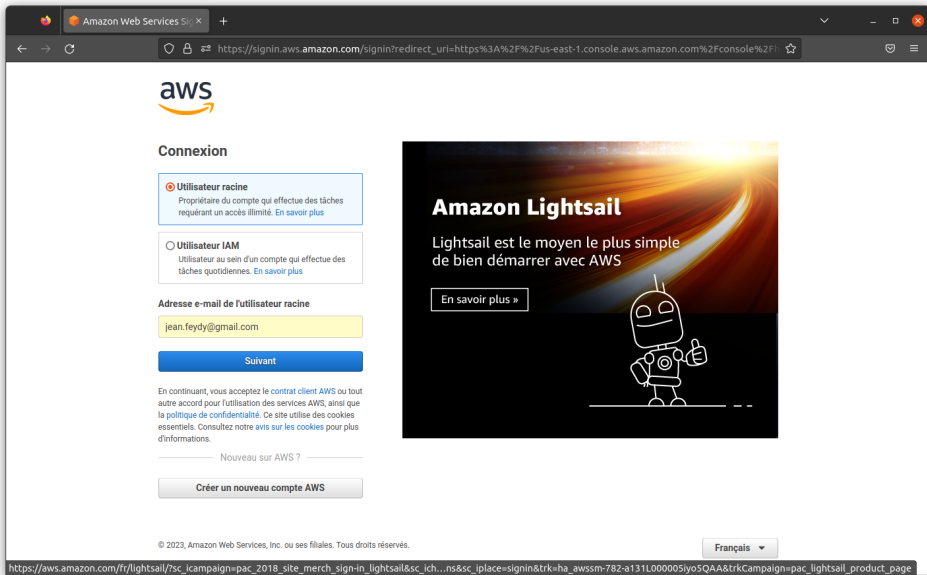
The terminal at the bottom shows the output of a command:

```
exlive/texmf-dist/fonts/opentype/adobe/sourcesanspro/SourceSansPro-Bold.otf>
Output written on slides.pdf (16 pages, 99102 bytes).
Transcript written on slides.log.
jfeidy@pl1174-pro:~/Documents/Talks/Rouen_2023$
```

A red circle highlights the gear icon in the bottom-left corner of the Visual Studio Code interface, which is used to access the settings menu.

Getting access to a cluster

Paperwork-free option: rent an instance on AWS EC2



The screenshot shows the AWS sign-in page in a browser. The browser's address bar shows the URL: `https://signin.aws.amazon.com/signin?redirect_uri=https%3A%2F%2Fus-east-1.console.aws.amazon.com%2Fconsole%2F...`. The page features the AWS logo at the top left. Below it, the heading "Connexion" is followed by two login options: "Utilisateur racine" (selected) and "Utilisateur IAM". The "Utilisateur racine" option includes a description: "Propriétaire du compte qui effectue des tâches requérant un accès illimité. [En savoir plus](#)". The "Utilisateur IAM" option includes: "Utilisateur au sein d'un compte qui effectue des tâches quotidiennes. [En savoir plus](#)". Below these options is a text input field for the email address, containing "jean.fedy@gmail.com", and a blue "Suivant" button. A paragraph of legal text follows, mentioning the AWS client agreement and privacy policy. At the bottom left, there is a link for "Nouveau sur AWS ?" and a button for "Créer un nouveau compte AWS". On the right side of the page, there is a large promotional banner for "Amazon Lightsail" with the text "Lightsail est le moyen le plus simple de bien démarrer avec AWS" and a "En savoir plus »" button. The banner also features a cartoon robot character. At the bottom right, there is a language selector set to "Français". The footer contains the copyright notice "© 2023, Amazon Web Services, Inc. ou ses filiales. Tous droits réservés." and a long URL.

aws

Connexion

☒ **Utilisateur racine**
Propriétaire du compte qui effectue des tâches requérant un accès illimité. [En savoir plus](#)

☐ **Utilisateur IAM**
Utilisateur au sein d'un compte qui effectue des tâches quotidiennes. [En savoir plus](#)

Adresse e-mail de l'utilisateur racine

jean.fedy@gmail.com

Suivant

En continuant, vous acceptez le [contrat client AWS](#) ou tout autre accord pour l'utilisation des services AWS, ainsi que la [politique de confidentialité](#). Ce site utilise des cookies essentiels. Consultez notre [avis sur les cookies](#) pour plus d'informations.

Nouveau sur AWS ?

Créer un nouveau compte AWS

Amazon Lightsail

Lightsail est le moyen le plus simple de bien démarrer avec AWS

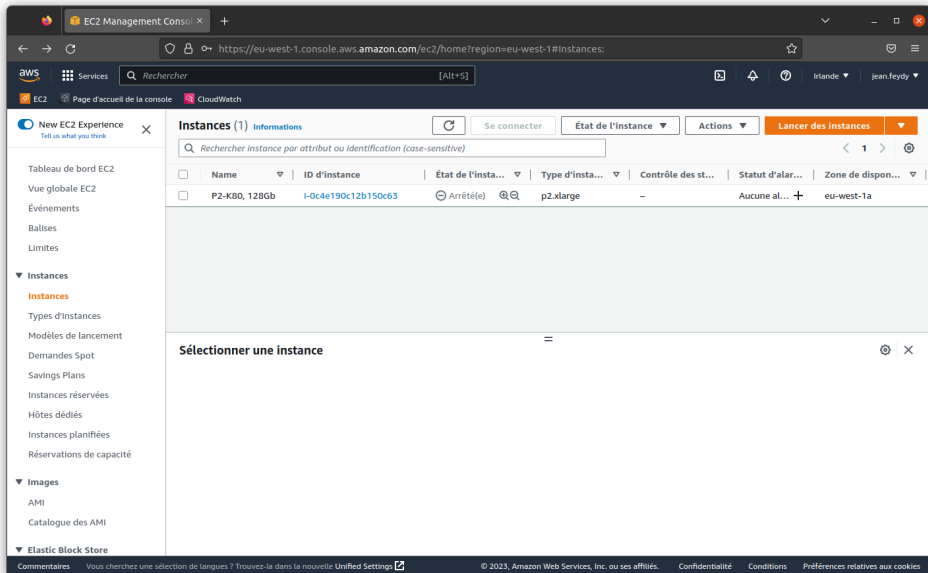
En savoir plus »

Français

© 2023, Amazon Web Services, Inc. ou ses filiales. Tous droits réservés.

`https://aws.amazon.com/fr/lightsail/?sc_icampaign=pac_2018_site_merch_sign-in_lightsail&sc_lch...ns&sc_lplace=signin&trk=ha_aws-sm-782-a1311000005iyo5QAA&trkCampaign=pac_lightsail_product_page`

Get full access to a custom configuration



The screenshot displays the AWS Management Console for the EC2 service in the eu-west-1 region. The left-hand navigation pane includes sections for 'New EC2 Experience', 'Tableau de bord EC2', 'Instances', 'Images', and 'Elastic Block Store'. The 'Instances' section is currently selected, showing a table with one instance: 'P2-K80, 128Gb' with ID 'I-0c4e190c12b150c63', which is in a 'Stopped' state. Below the table, a modal window titled 'Sélectionner une instance' is open. The top of the console features a search bar, navigation buttons, and a user profile. The bottom of the page contains a footer with legal information and a language selection link.

EC2 Management Console

Services Rechercher [Alt+S]

EC2 Page d'accueil de la console CloudWatch

New EC2 Experience Tell us what you think

Tableau de bord EC2

Vue globale EC2

Événements

Balises

Limites

▼ Instances

Instances

Types d'instances

Modèles de lancement

Demandes Spot

Savings Plans

Instances réservées

Hôtes dédiés

Instances planifiées

Réservations de capacité

▼ Images

AMI

Catalogue des AMI

▼ Elastic Block Store

Instances (1) Informations

Se connecter État de l'instance Actions Lancer des instances

Rechercher instance par attribut ou identification (case-sensitive)

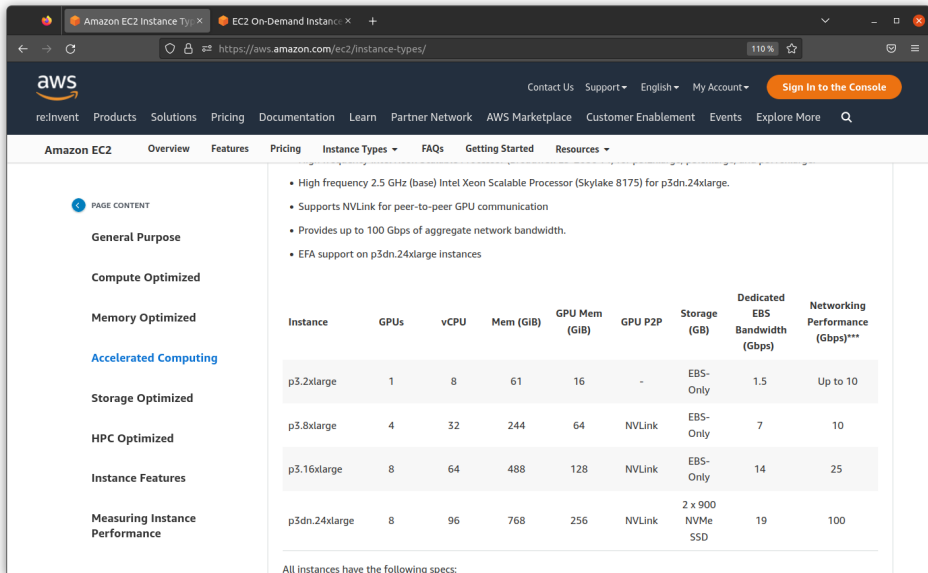
	Name	ID d'instance	État de l'instance	Type d'instance	Contrôle des st...	Statut d'alar...	Zone de dispon...
<input type="checkbox"/>	P2-K80, 128Gb	I-0c4e190c12b150c63	Arrêté(e)	p2.xlarge	-	Aucune al... +	eu-west-1a

Sélectionner une instance

Commentaires Vous cherchez une sélection de langues ? Trouvez-la dans la nouvelle Unified Settings

© 2023, Amazon Web Services, Inc. ou ses affiliés. Confidentialité Conditions Préférences relatives aux cookies

All types of hardware are available



The screenshot shows the AWS EC2 Instance Types page. The left sidebar contains a navigation menu with the following items: General Purpose, Compute Optimized, Memory Optimized, Accelerated Computing (highlighted in blue), Storage Optimized, HPC Optimized, Instance Features, and Measuring Instance Performance. The main content area displays a list of instance types with their specifications. The table includes columns for Instance, GPUs, vCPU, Mem (GiB), GPU Mem (GiB), GPU P2P, Storage (GB), Dedicated EBS Bandwidth (Gbps), and Networking Performance (Gbps). The instance types listed are p3.2xlarge, p3.8xlarge, p3.16xlarge, and p3dn.24xlarge. The p3dn.24xlarge instance has a storage specification of 2 x 900 NVMe SSD.

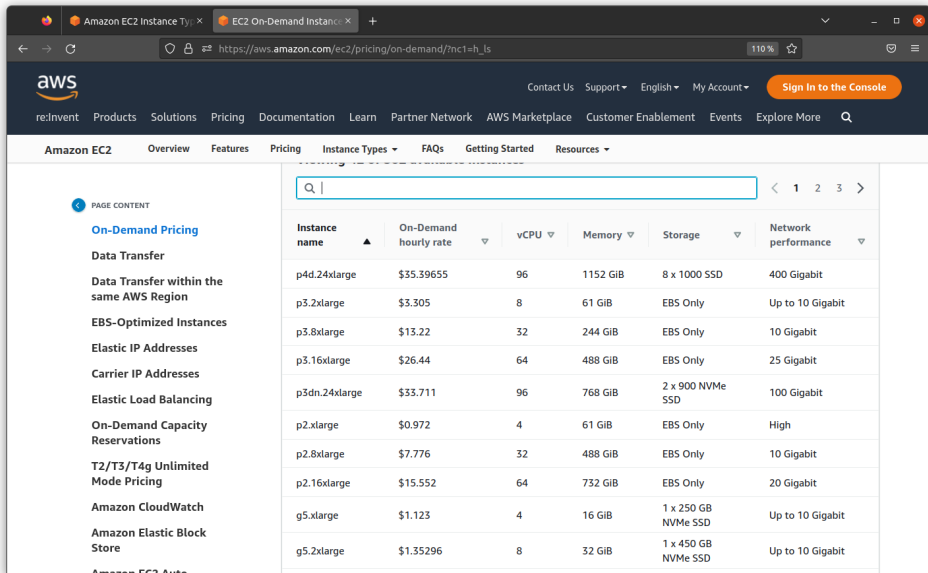
High frequency 2.5 GHz (base) Intel Xeon Scalable Processor (Skylake 8175) for p3dn.24xlarge.

- Supports NVLink for peer-to-peer GPU communication
- Provides up to 100 Gbps of aggregate network bandwidth.
- EFA support on p3dn.24xlarge instances

Instance	GPUs	vCPU	Mem (GiB)	GPU Mem (GiB)	GPU P2P	Storage (GB)	Dedicated EBS Bandwidth (Gbps)	Networking Performance (Gbps)***
p3.2xlarge	1	8	61	16	-	EBS-Only	1.5	Up to 10
p3.8xlarge	4	32	244	64	NVLink	EBS-Only	7	10
p3.16xlarge	8	64	488	128	NVLink	EBS-Only	14	25
p3dn.24xlarge	8	96	768	256	NVLink	2 x 900 NVMe SSD	19	100

All instances have the following specs:

Typical prices: about 3€ per hour for a high-end GPU



The screenshot shows the AWS EC2 On-Demand Pricing page. The left sidebar contains a 'PAGE CONTENT' menu with links to various services and pricing models. The main content area displays a table of instance types with columns for Instance name, On-Demand hourly rate, vCPU, Memory, Storage, and Network performance. The table lists various instance types, including p4d.24xlarge, p3.2xlarge, p3.8xlarge, p3.16xlarge, p3dn.24xlarge, p2.xlarge, p2.8xlarge, p2.16xlarge, g5.xlarge, and g5.2xlarge, along with their respective hourly rates and specifications.

Instance name	On-Demand hourly rate	vCPU	Memory	Storage	Network performance
p4d.24xlarge	\$35.39655	96	1152 GiB	8 x 1000 SSD	400 Gigabit
p3.2xlarge	\$3.305	8	61 GiB	EBS Only	Up to 10 Gigabit
p3.8xlarge	\$13.22	32	244 GiB	EBS Only	10 Gigabit
p3.16xlarge	\$26.44	64	488 GiB	EBS Only	25 Gigabit
p3dn.24xlarge	\$33.711	96	768 GiB	2 x 900 NVMe SSD	100 Gigabit
p2.xlarge	\$0.972	4	61 GiB	EBS Only	High
p2.8xlarge	\$7.776	32	488 GiB	EBS Only	10 Gigabit
p2.16xlarge	\$15.552	64	732 GiB	EBS Only	20 Gigabit
g5.xlarge	\$1.123	4	16 GiB	1 x 250 GB NVMe SSD	Up to 10 Gigabit
g5.2xlarge	\$1.35296	8	32 GiB	1 x 450 GB NVMe SSD	Up to 10 Gigabit

What about INRIA?

Commercial cloud services:

- Great for **reproducible benchmarks**.
- A “**marché public**” for AWS, GCP, OVH is currently being approved.
- Until then: pay in advance from your **own pocket**,
get a re-fund from your **research grants** as “menus frais” for up to **500€ / month**.

INRIA clusters:

- Local cluster at INRIA Paris: Cleps.
- Currently being merged within the Grid5k framework.

National cluster for AI research:

- Jean Zay: lots of **high-end GPUs** and **dedicated support**, free of charge.
- Main drawback: super strict on **security**.

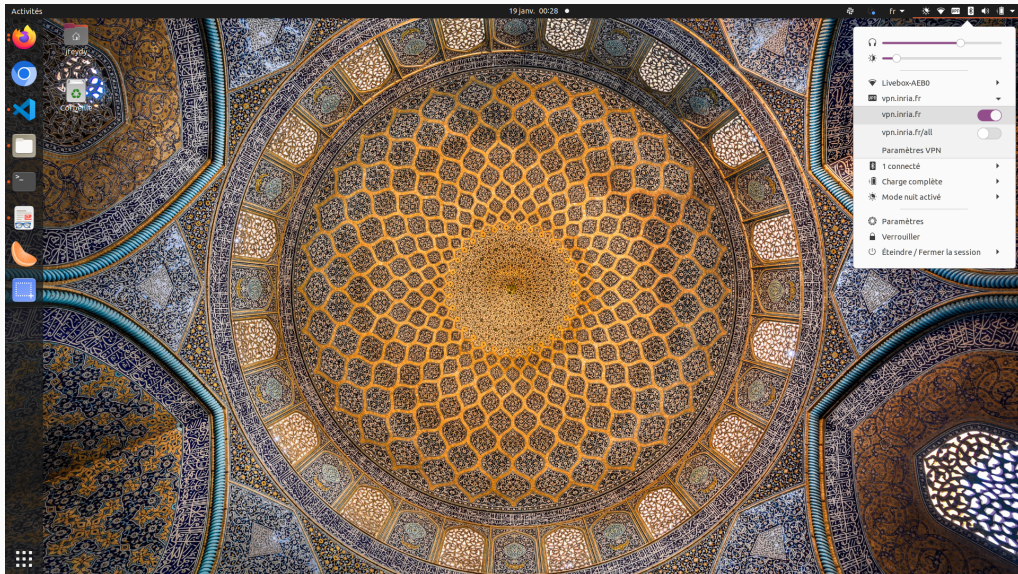
Full documentation is available:

<https://paris-cluster-2019.gitlabpages.inria.fr/cleps/cleps-userguide/index.html>

Three conditions:

- Have a `jfeydy@inria.fr` account.
- Use your **professional** computer.
- Connect to the **INRIA network** with a VPN:
`sudo openconnect -u jfeydy vpn.inria.fr`

On INRIA laptops: just use the graphical VPN menu



SSH config: ~/.ssh/config or %userprofile%\ssh\config

```
1  # Dummy local connection:
2  Host localhost
3  HostName 127.0.0.1
4  User jean
5
6  # Cleps - Inria Paris:
7  Host cleps
8  User jfeydy
9  HostName cleps.inria.fr
10 port 22
11 ControlMaster auto
12 ForwardX11 no
13 ForwardAgent yes
```

ssh cleps

[illegible]

The Slurm manager: launch scripts with `sbatch myproject.batch`

```
1  #!/bin/bash
2  #SBATCH --job-name=myproject      # create a short name for your job
3  #SBATCH --mail-type=ALL           # mail events: NONE, BEGIN, END, FAIL...
4  #SBATCH --mail-user=jean.feydy@inria.fr
5  #SBATCH --nodes=1                 # node count
6  #SBATCH --ntasks=1                # total number of tasks across all nodes
7  #SBATCH --cpus-per-task=16        # cpu-cores per task
8  #SBATCH --partition=gpu           # Name of the partition
9  #SBATCH --gres=gpu:rtx6000:1      # Use a GPU node
10 #SBATCH --mem=40G                  # Total memory allocated
11 #SBATCH --hint=multithread         # we get physical cores not logical
12 #SBATCH --time=03:00:00           # total run time limit (HH:MM:SS)
13 #SBATCH --output=logs/myproject.out # output file name
14
15 echo "### Running $SLURM_JOB_NAME ###"
16 cd ${SLURM_SUBMIT_DIR}
17 your bash script...
```

Ideal for AI research on non-sensitive data:

- **90k** high-end CPU cores with **450 TB of RAM**.
- **2,700 V100** GPUs + **416 A100** GPUs.
- **Admins** who understand the needs of AI/ML researchers: see e.g.
<http://www.idris.fr/jean-zay/pre-post/jean-zay-jupyter-notebook.html>

Full documentation is available:

- Paperwork and tutorials: <https://jean-zay-doc.readthedocs.io/>
- Technical documentation: <http://www.idris.fr/jean-zay/>

Getting access to Jean Zay

My main steps – Summer 2022:

1. **Follow** <https://jean-zay-doc.readthedocs.io/en/latest/access-procedure/>
2. **Asked** for 8,000 V100 hours + 2,000 A100 hours per year.
3. Ask Eric Fleury to sign the “**Visa request**”
via his assistant `celine.scherschen@inria.fr`.
4. Ask a “**security stamp**” to `Laurent.Le_Pendeven@inria.fr`.
5. Tell Jean Zay that I will connect from the **INRIA IP address**:
`ssh.paris.inria.fr = 128.93.96.2`.
6. Create a “ticket” on the INRIA system to add my SSH `id_rsa.pub`
to the **list of known users** on `ssh.paris.inria.fr`.
Dedicated category: “Demande d’accès distants serveurs : Accès Bastion SSH”.

SSH config: ~/.ssh/config or %userprofile%\ssh\config

```
14  # Inria bastion for Jean Zay:
15  Host inria-ssh
16  User jfeydy
17  HostName ssh.paris.inria.fr
18
19  # Jean Zay:
20  Host jean-zay
21  User my_personal_id
22  HostName jean-zay.idris.fr
23  ProxyJump inria-ssh
24
25
26
```


Set up a reproducible environment

Virtual environment with Python – to handle conflicts:

- **pip**: for pure Python packages.
- **conda**: to handle C/C++/binary extensions.

Containers – fast and lightweight virtual machines to ensure portability:

- **Docker**: industry standard.
- **Singularity**: standard on shared academic clusters, more secure but more rigid, 100% compatible with Docker.

Install Singularity from <https://github.com/sylabs/singularity/releases> or use a command such as “`module load singularity`” (on a cluster).

Full documentation is available online. You may:

- Define precisely **your own configuration** with a bash install script.
- Keep your environment clean with a custom “Home” folder per project.
This is especially important to ensure **full reproducibility**.
- Connect to your containers via SSH to benefit from VSCode integration, etc.

Install a configuration that you like from DockerHub

```
14 # For large images, the in-memory tmpfs may be too small:
15 # we create a folder on the hard drive instead.
16 mkdir tmp
17 # Perform a virtual install of the official KeOps environment
18 # (Ubuntu + CUDA + PyTorch + KeOps + survival-GPU + ...)
19 # in the immutable file image.sif (~6 GB):
20 SINGULARITY_TMPDIR=`pwd`/tmp \
21 singularity build image.sif docker://getkeops/keops-full:latest
22
23 # Note that this requires admin rights and may take some time:
24 # you may perform this step on your laptop, and copy-paste
25 # image.sif on the cluster afterwards.
26
27 # Then, run a shell in the virtual machine with NVidia GPU support:
28 singularity shell --nv image.sif
```

Please feel free to ask me questions anytime!

