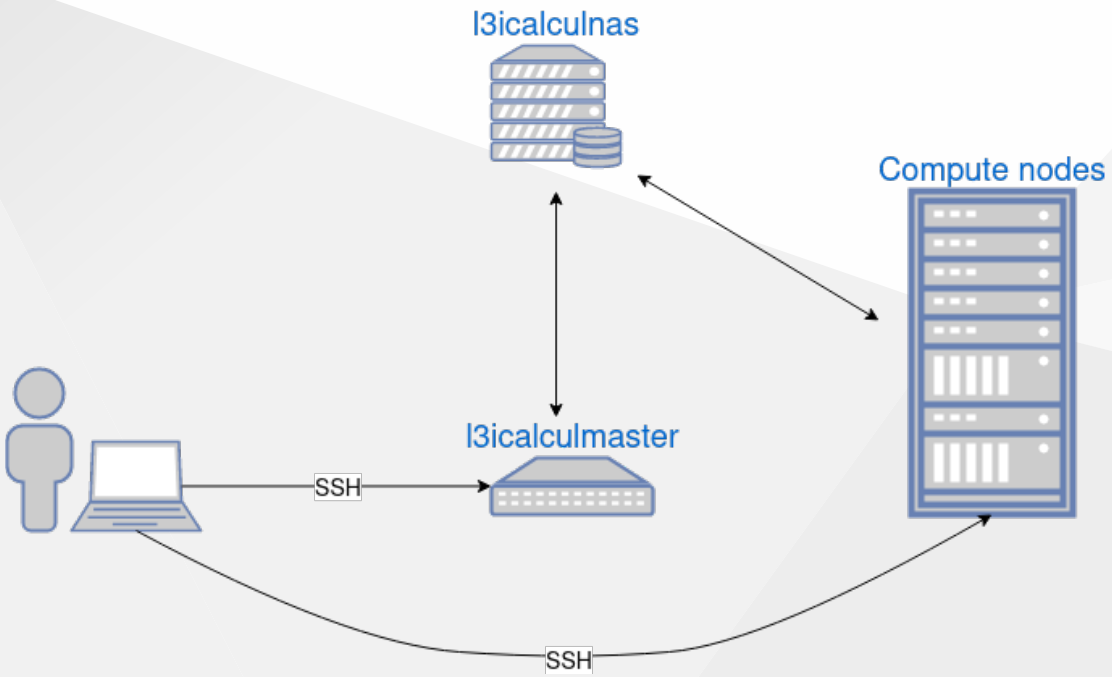
L3iCalcul

SLURM

Simple Linux Utility for Resource Management

https://slurm.schedmd.com/

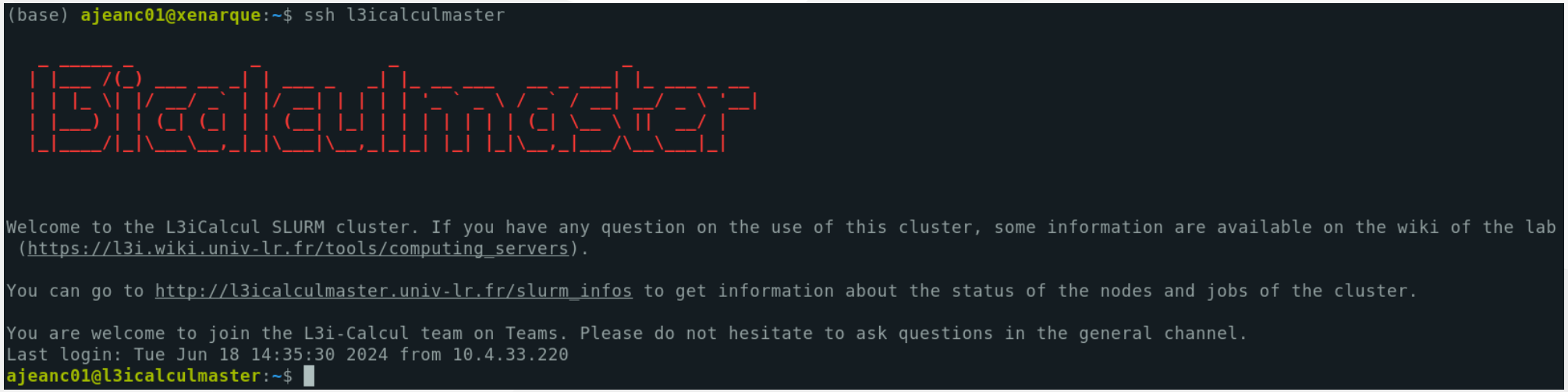


Connect to the master node

Network: ULR-secure or VPN

```
ssh user@l3icalculmaster
```

Connect using your LDAP credentials.



Concepts

- **Job:** allocation of computing resources, batch script
- **Job step:** `srun` inside a batch script, sequential or parallel execution, (default steps: *batch* and *extern*)
- **Task:** CPUs are allocated per task

sbatch parameters (1/2)

- **--partition:** the partition you want your job to run onto.
- **--time:** the maximum time of your job. A default time of 7 days is set on every partition.
- **--job-name:** a name to give to your job
- **--error:** the file where the standard error (stderr) of your job will be written.
- **--output:** the file where the standard output (stdout) of your job will be written. If no `--error` option is set, the standard error will also be written to this file.

sbatch parameters (2/2)

- **--nodes:** the number of nodes you want your job to run onto.
- **--mem:** the memory allocated per node requested. You can also use the `--mem-per-cpu` and `--mem-per-gpu` options. These three options are mutually exclusive. You can specify different units
- **--gres:** allows you to request for GPUs. The value of this parameter is `gpu:n` where `n` is the number of GPU you want.
- **--cpus-per-task:** indicate the number of cpus your tasks will use. If you omit this parameter, you'll be given 1 or 2 cores per task. Do not forget to double check that your code is actually using the CPUs you requested.

Python virtual environment

To avoid conflicts with dependencies you must use a virtual environment. `Anaconda` is installed on every node and is configured to store your virtual environments in your home directory (shared among nodes).

```
module load Anaconda
```

```
source /opt/easybuild/software/Anaconda3/2024.02-1/etc/profile.d/conda.sh
```

Anaconda

- `conda env list` to list your previously created environments
- `conda create -n <my_env> python=3.11 -y` creates a virtual environment
- `conda activate <my_env>` sets the PATH variable to map the Python binary
- `conda deactivate` restore the system environment

Install dependencies

```
conda activate <my_env>
```

```
pip install -r requirements.txt
```

or

```
pip install torch numpy...
```

CALI3

- First iteration of the DOREMI regional datacenter for scientific research
- Free access for members of the lab
- general wiki : https://redmine.mcia.fr/projects/cluster-cali3/wiki
- registration : https://redmine.mcia.fr/projects/cluster-cali3/wiki/Compte

CALI3 - Registration

- Create a MCIA account : https://account.mcia.fr/requests/signup
- Request access to CALI3
- Add public ssh key to your profile
- Connect to the cluster `ssh <username>@cali3.unilim.fr`

CURTA

- A lot of CPU cores available
- User guide : https://redmine.mcia.fr/projects/cluster-curta/wiki/Guide_de_l'utilisateur

Final thoughts

- Get information and help
  - Wiki : https://l3i.wiki.univ-lr.fr/tools/computing_servers
  - L3i-Calcul on Teams
  - Monthly meeting in room 101 (last wednesday of the month at 11 A.M.)

Final thoughts

- Data on SLURM (`/Utilisateurs` and `/data`) is **NOT** backed up
- You can use one of the storage solutions provided
- http://l3i-services.univ-lr.fr/