# **SCITAS Computing Guide**

## **Connecting to the cluster**

Use the following command to connect to the cluster.

ssh <username>@izar.epfl.ch

Locate the shared storage: scratch

cd /scratch/izar/<username>

This is a high-performance temporary space, not backed up, low redundancy, built for performance, and local to each cluster. **The automatic cleanup procedure deletes files (older than 2 weeks) without warning (when occupancy reaches a threshold)!** **The storage is intended for disposable files: intermediary results, temporary files!**

The /home/<username> directory is a shared storage available on all clusters and backed up to a remote site. It has a per-user quota of 100 GB. **You can use this for important files: source code, final results, checkpoints, etc.**

## **Data transfer**

*# To the cluster*

scp my-script.py <username>@izar:./code/

rsync -a ./dataset <username>@izar:/scratch/izar/<username>/

*# From the cluster*

scp -r <username>@izar:/scratch/izar/<username>/results .

rsync -a <username>@izar:./updated-code/ Documents/updated-code

If you are new to scp or rsync, you can check this [introduction](https://contabo.com/blog/how-to-use-rsync-and-scp-to-transfer-files-on-a-vps/#:~:text=Rsync%20and%20SCP%2C%20both%20harnessed,solution%20for%20secure%20file%20transfers.)

## **Submitting a job**

### **Batch Systems**

1. Interactive: You sit in front of your computer, open Jupyter or VScode on the cluster, and work
2. You put the workflow into a script and put it in a queue. Your jobs will run when appropriate resources become available.

### **Slurm Scheduler**

The cluster uses Slurm to manage and schedule job submissions. Here is a great [tutorial](https://github.com/abalter/slurm-tutorial/wiki) for Slurm with Machine Learning.

sbatch: submits jobs to the batch system. A typical workflow to get your computation done is:

**1: Create a short job-script**

Example:

#!/bin/bash -l

*#SBATCH --chdir /scratch/izar/<put-your-username-here>*

*#SBATCH --ntasks 1*

*#SBATCH --cpus-per-task 1*

*#SBATCH --mem 4G*

*#SBATCH --time 2:00*

*#SBATCH --gres gpu:1*

*#SBATCH --account cs-552*

*#SBATCH --qos cs-552*

*#SBATCH --reservation cs-552*

*# `SBATCH --something` is how you tell SLURM what resources you need*

*# The --reservation line only works during the 2-week period*

*# where 80 GPUs are available. Remove otherwise*

echo "hello from $(hostname)"

sleep 2

* --ntasks 1: the number of MPI tasks per job. If not specified, the default is 1.
* --cpu-per-task 1: the number of CPUs per task for multithread applications. If not specified, the default is 1. Note that this value cannot be more than the number of cores/cpus in a compute node!
* --mem 4G: the required memory per node. If not specified the default is 4096 MB per core. Don’t ask for the exact amount on a node (192 GB). The OS also needs some.
* --time 01:00:00 or 2-23:00:00: how long will your job run for with the format day-hour:min:second. Choose the time wisely. Small jobs fit in the gaps between big jobs.

**2: Submit the script to the batch system**

$ sbatch ex1.sh

Submitted batch job 1509281

*# !! The number returned by sbatch is your Job ID to uniquely identify your task.*

*# Remembr this Job ID.*

$ cat /scratch/izar/<username>/slurm-1509281.out

hello from i02

To check your job's status in the queues, run the following commands:

*# To check your own jobs*

squeue -u <username>

*# To check just your pending (PD) jobs:*

squeue -u <username> -t PD

To cancel a job, use the following commands:

*# Cancel a specific job*

scancel <JOB\_ID>

*# Cancel all your jobs*

scancel -u <username>

*# Cancel all your jobs that are not yet running*

scancel -u <username> -t PENDING

## **Interactive Access**

For debugging or running applications interactively, we don’t want to submit a batch job.

You can use the command Sinteract to start an interactive session. For example:

[<user>@izar ~]$ Sinteract -a cs-552 -q cs-552 -g gpu:1 -t 00:05:00 -c 20 -m 90000

Cores: 20

Tasks: 1

Time: 00:05:00

Memory: 90000

Partition: gpu

Account: cs-552

Jobname: interact

Resource: gpu:1

QOS: cs-552

Reservation:

Constraints:

salloc: Pending job allocation 2050717

salloc: job 2050717 queued and waiting for resources

salloc: job 2050717 has been allocated resources

salloc: Granted job allocation 2050717

salloc: Nodes i01 are ready for job

Waiting for X11 setup...

[<user>@i01 ~]$

In interactive sessions, cores, RAM, and GPUs are reserved for you, whether being used or not. You must close your session once done to free resources for others!

Note that The debug partition/queue is available to everyone. For the course, you can use Sinteract -a cs-552 -q cs-552 -g gpu:1

For an advanced tutorial on how to use Jupyter Notebook remotely with the interactive session, please take a look at this [user document](https://scitas-doc.epfl.ch/advanced-guide/python/jupyter/) provided by the SCITAS team.

## **Create a Python Virtual Environment**

To start working in your interactive session or to execute a job, you should create a virtual environment as your workspace:

module load gcc python

virtualenv --system-site-packages ~/venvs/course\_py-3.10

You can use this new virtual environment anytime by activating it:

source ~/venvs/course\_py-3.10/bin/activate

*# Now you can run any pip and python commands in this env*

pip install -r requirements.txt

python3 my\_script.py

To stop using the virtual environment, simply run: deactivate

## **Simple test run**

We provide you a [testing script](https://drive.google.com/file/d/1Gf9vk4oKXP6s13CEkcT4mXZToRPzckrK/view?usp=sharing) with a small training loop requiring Pytorch and GPU. You can run this script to check if the setup with Pytorch installation and GPU support works as expected.

## **How to get help?**

If you ever run into issues, you can reach the support team for help:

1. Send a mail to [1234@epfl.ch](about:blank)
2. Start the subject with HPC
3. Organize with the TAs! They likely know the answer to basic problems.

provide as many of the following as possible:

1. The Job ID.
2. The directory location and name of the submission script.
3. Where the “slurm-\*.out” file is to be found.
4. How the “sbatch” command was used to submit it.
5. The output from “env” and “module list” commands.

## **Useful Links and References**

* [SCITAS web site](http://scitas.epfl.ch/)
* [SCITAS courses (Intro to Linux, Clusters)](https://www.epfl.ch/research/facilities/scitas/documentation/training/)
* [(in particular) SCITAS documentation space](http://scitas-doc.epfl.ch/)
* [SLURM man pages](http://slurm.schedmd.com/man_index.html)