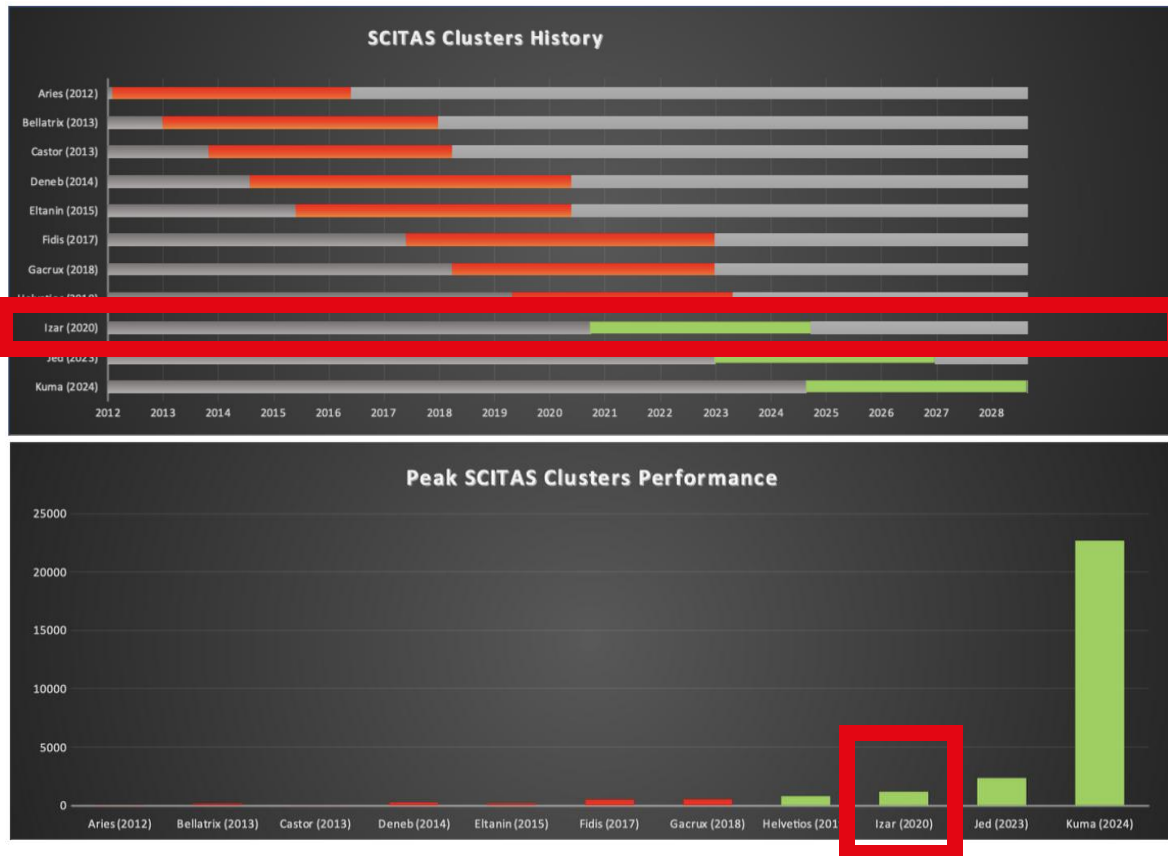


Using SCITAS

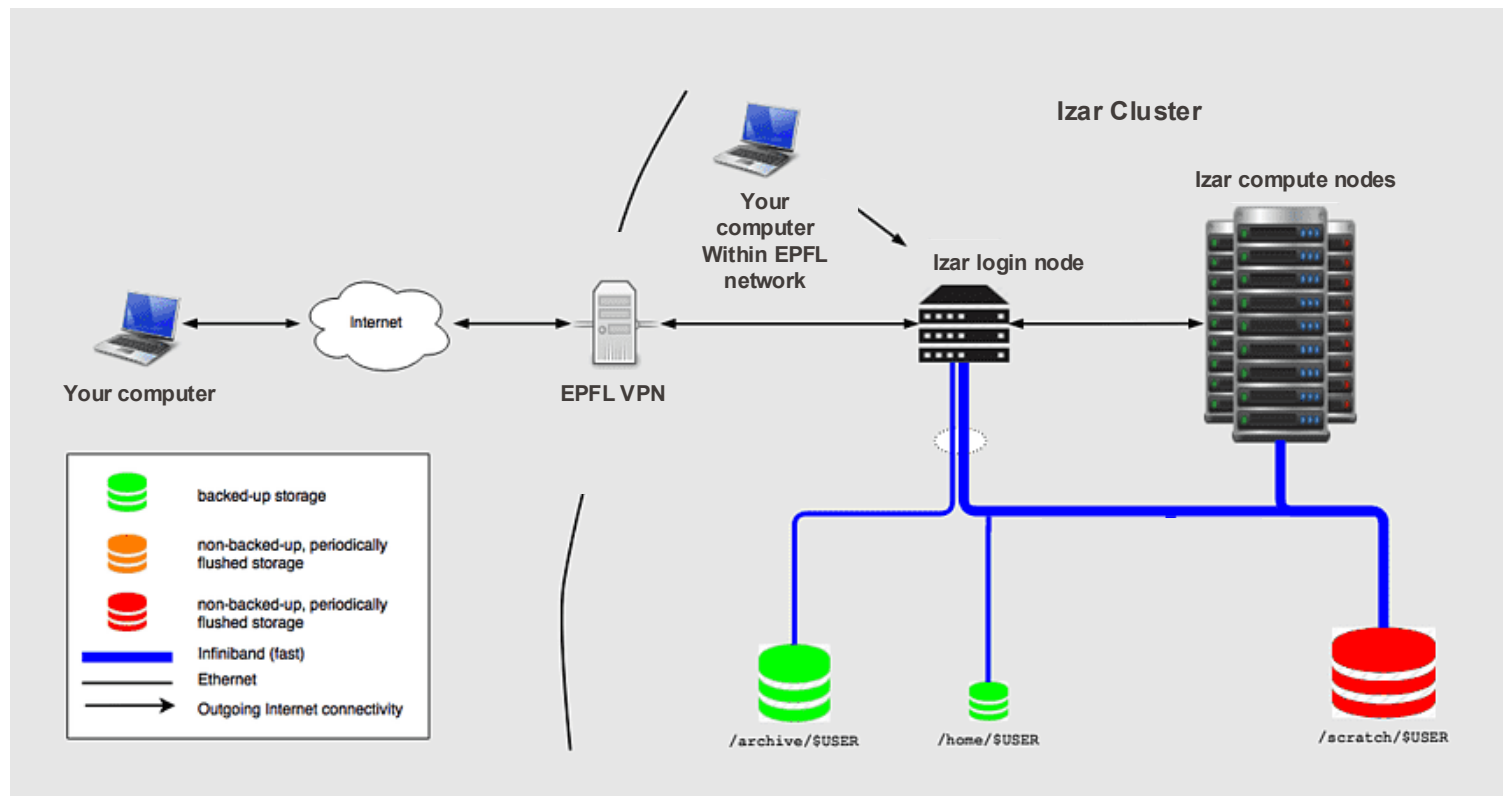
A guide to first steps

ENV-540

Jonathan Sauder



SCITAS manages GPU and CPU clusters. You will be using the izar GPU cluster.



Step 1: Connecting to Izar

`ssh <user>@izar.epfl.ch` will give you a shell on the login node

```

jonathan@eceoh219709 ~ % ssh sauder@izar.epfl.ch
Last login: Mon Nov  4 16:18:33 2024 from vpn-253-055.epfl.ch
##### Welcome to IZAR #####
https://go.epfl.ch/izar-spec

Access to the computational resources on this cluster is restricted to
Master students and course attendees who have submitted a request.

----- Tips -----
Do not wait for the cluster load to decrease to submit jobs.
- Submit jobs to HPC batch-schedulers without waiting for
cluster load to decrease.
- Jobs will be queued and run when system conditions allow.
https://scitas-doc.epfl.ch/user-guide/using-clusters/slurm-job-priorities/#checking-job-priority

Happy computing! 🍷🚀

----- Announcements -----

General
2024-09-04:
🔴 Izar cluster has reached its end of life and will be
repurposed to education starting November 1st, 2024.
Pay-per-use users can move to our new GPU cluster Kuma
(kuma.hpc.epfl.ch) right now.

More information:
https://scitas-doc.epfl.ch/blog/2024/09/02/kuma-beta-opening/

System
2024-03-01:
Cost estimation when running job is not correct (using
sbatch/srun/salloc).
The Sausage tool is working as expected (sausage --help)

----- Pricing -----

GPU: CHF 0.0000/gpu/hour

----- Sausage -----

USERNAME : sauder
Global usage from 2024-11-01 to 2024-11-30

No data to display

sausage v0.12.1.2

----- Infos -----
Documentation ..... https://go.epfl.ch/scitas-doc
Usage statistics ..... https://go.epfl.ch/scitas-stats
Terms and conditions..... https://go.epfl.ch/scitas-terms
Need Help? Questions? Comments? 1234@epfl.ch

hpc_mtd v0.12.0
[sauder@izar1 ~]$

```

Step 2: Making your Python Environment

First, load Python (and CUDA/CUDNN):

```
module load gcc python
```

Then, (only the first time) start a new virtual environment:

```
python -m venv ipeo_venv
```

```
source ipeo_env/bin/activate
```

```
pip install <your modules, e.g. jupyter, torch, numpy>
```

After that, you can always reactivate your environment:

```
source ipeo_env/bin/activate
```

Step 3: Slurm Jobs - sinfo

sinfo

```
(ipeo_venv) [sauder@izar1 ~]$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
gpu*      up    infinite    4  drain* i[22,37-38,59]
gpu*      up    infinite    1   drng  i21
gpu*      up    infinite    9   mix  i[19,27-30,33-35,65]
gpu*      up    infinite    4  alloc  i[31-32],ixl[01-02]
gpu*      up    infinite   51  idle  i[01-18,20,23-26,36,39-58,60-62,66-69]
debug     up    infinite    1  idle  i63
build     up    infinite    1  idle  i64
test      up    infinite    1  idle  i70
gpu-xl    up    infinite    2  alloc  ixl[01-02]
```

Step 3: Slurm Jobs - Sinteract

```
Sinteract -p gpu -a env540 -g gpu:1 -c 32 -m 32G -t 00:20:00
```

```
nvidia-smi
```

```
[(ipeo_venv) [sauder@izar1 ~]$ Sinteract -p gpu -a env540 -g gpu:1 -t 00:20:00
Cores:      1
Tasks:      1
Time:       00:20:00
Memory:     4G
Partition:  gpu
Account:    env540
Jobname:    interact
Resource:   gpu:1
QOS:        gpu
Reservation:
Constraints:
```

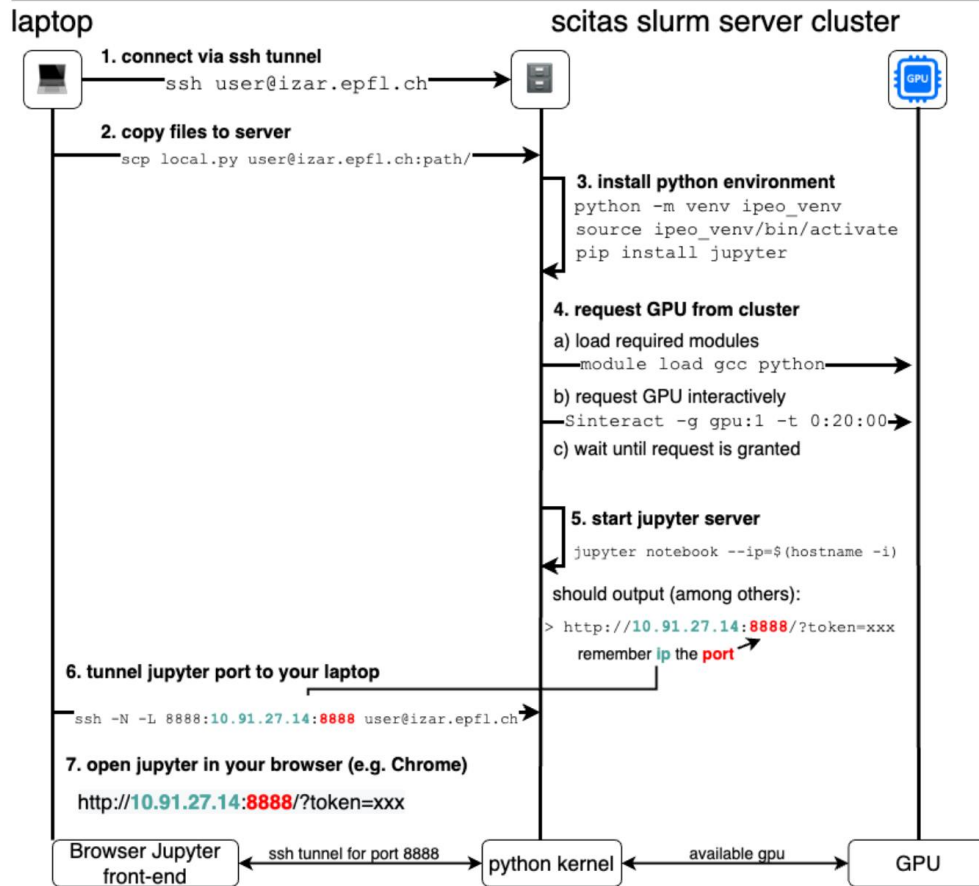
```
salloc: [ESTIMATION] The estimated cost of this job is CHF 0.00
salloc: Pending job allocation 2170530
salloc: job 2170530 queued and waiting for resources
salloc: job 2170530 has been allocated resources
salloc: Granted job allocation 2170530
salloc: Waiting for resource configuration
salloc: Nodes i29 are ready for job
Waiting for X11 setup..
[[sauder@i29 ~]$ nvidia-smi
Fri Nov  8 09:02:52 2024
```

```
-----
| NVIDIA-SMI 535.154.05                Driver Version: 535.154.05   CUDA Version: 12.2   |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----|
| GPU   Name                               Persistence-M   Bus-Id        Disp.A   Volatile Uncorr. ECC   |
| Fan  Temp  Perf              Pwr:Usage/Cap     |      Bus-Id        Memory-Usage  GPU-Util  Compute M. |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----|
|  0   Tesla V100-PCIe-32GB         On              00000000:08:00.0 Off      |          0%      Off      |
| N/A   29C    P0              23W / 250W     |          0MiB / 32768MiB      |          0%      Default |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----|
|                                                                              N/A          |
|                                                                              |
```

```
-----
| Processes:                               |
| GPU   GI   CI        PID   Type   Process name                               | GPU Memory |
| ID   ID                               |              |          Usage                               |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----|
|                                                                              | No running processes found |
|                                                                              |
```

32 CPU cores, 32 GB Memory

General Workflow



Detailed instructions for how to run jupyter notebooks in interactive sessions on Moodle!

Step 3: Slurm Jobs - sbatch

Submit your job on slurm. For this we write a run script, called 'job.run`:

```
#!/bin/bash -l
```

```
#SBATCH --account env540
```

```
#SBATCH --nodes=1
```

```
#SBATCH --ntasks=1
```

```
#SBATCH --cpus-per-task=32
```

```
#SBATCH --partition=gpu
```

```
#SBATCH --qos=gpu
```

```
#SBATCH --gres=gpu:1
```

```
#SBATCH --time=00:01:00
```

```
# Load modules
```

```
module load gcc python
```

```
# Activate virtual environment or conda environment
```

```
source ~/ipeo_venv/bin/activate # Replace with your environment setup
```

```
# Run your Python script
```

```
python main.py # Replace with your script name
```

Step 3: Slurm Jobs - sbatch

```
sbatch one_gpu_training.run
```

```
[(ipeo_venv) [sauder@izar1 ~]$ sbatch one_gpu_training.run  
sbatch: [ESTIMATION] The estimated cost of this job is CHF 0.00  
Submitted batch job 2170532
```

This will create a file called `slurm-2170532.out`, where the stdout from the main.py file will be written

Slurm directives can also be given in the command line, superseding what you set on the script itself:

```
sbatch --time=2-00:00:00 job.run
```

would ask for a 2 day time limit, regardless of the 1 minute limit set in the script. The longer the expected time, the lower the job priority!

squeue

```
[(ipeco_venv) [sauder@izar1 ~]$ squeue
  JOBID PARTITION   NAME     USER  ST       TIME  NODES NODELIST(REASON)
  2157651      gpu trial.sh  nvarini PD      0:00      1 (QOSMinGRES)
  2168733      gpu hello.ru giugiaro PD      0:00      1 (QOSMinGRES)
  2154558      gpu eval.sh  saillen PD      0:00      1 (QOSMinGRES)
  2154557      gpu run_open saillen PD      0:00      1 (QOSMinGRES)
  2154486      gpu run_open saillen PD      0:00      1 (QOSMinGRES)
  2154713      gpu brain_ma bocini  PD      0:00      1 (QOSMinGRES)
  2154306      gpu cryospar lpdi-cry PD      0:00      1 (QOSMinGRES)
  2154305      gpu cryospar lpdi-cry PD      0:00      1 (QOSMinGRES)
  2154304      gpu cryospar lpdi-cry PD      0:00      1 (QOSMinGRES)
  2154303      gpu cryospar lpdi-cry PD      0:00      1 (QOSMinGRES)
  2154302      gpu cryospar lpdi-cry PD      0:00      1 (QOSMinGRES)
  2154299      gpu cryospar lpdi-cry PD      0:00      1 (QOSMinGRES)
  2154294      gpu cryospar lpdi-cry PD      0:00      1 (QOSMinGRES)
  2154288      gpu cryospar lpdi-cry PD      0:00      1 (QOSMinGRES)
  2154287      gpu cryospar lpdi-cry PD      0:00      1 (QOSMinGRES)
  2154286      gpu cryospar lpdi-cry PD      0:00      1 (QOSMinGRES)
  2154284      gpu cryospar lpdi-cry PD      0:00      1 (QOSMinGRES)
  2154283      gpu cryospar lpdi-cry PD      0:00      1 (QOSMinGRES)
  2170193      gpu interact goverde  R  1-18:38:09  1 i28
  2170192      gpu interact goverde  R  1-18:45:50  1 i19
  2170180      gpu interact goverde  R  1-19:17:12  1 i35
  2169990      gpu train.ru  ymiao  R  2-17:22:24  1 i21
  2169934      gpu interact pguo    R  2-21:36:48  1 i19
  2170520      gpu interact orliac  R    1:09:50     1 i31
  2170406      gpu clip_cc3 bashir  R   15:57:25   1 ix102
  2170410      gpu clip_cc3 bashir  R   15:40:34   1 ix101
  2170497      gpu invarian moutarli R    8:51:13     1 i33
  2170416      gpu clip_cc3 bashir  R   14:43:52   1 i65
  2170334      gpu da_Li     yxu    R   18:17:55     1 i32
  2170286      gpu train_im ybecker R   21:31:54     1 i28
  2170422      gpu CCVPE_os qngo   R   13:50:50     1 i29
  2170419      gpu CCVPE_os qngo   R   13:56:34     1 i27
  2170521      gpu eval_ran jurcut  R    1:04:59     1 i34
  2170515      gpu train_gv ymiao  R    2:46:00     1 i30
  2170428      gpu interact pguo    R   13:38:00     1 i27
```

`squeue -u <username>` to see only your jobs!

Step 3: Slurm Jobs - scancel

To cancel a specific job:

```
scancel <job_id>
```

To cancel all your jobs (use with care!):

```
scancel -u $USER
```

To cancel all your jobs that are not yet running:

```
scancel -u $USER -t PENDING
```

General tips:

- There is a default maximum time for a job, if you request more time, your job will never be executed
- Slurm is well-documented and used in almost all clusters worldwide
- This is not an exhaustive tutorial, but covers the basic needs

- Refer to the SCITAS documentation at:
<https://scitas-doc.epfl.ch/>

- We uploaded instructions on how to launch a jupyter notebook on SCITAS