**CLUSTER COMPUTING STUFF**

SIMPLE THAT NO ONE SAID

* activate uni vpn
  + nrKBH0MI5fcn&X\*x
* ssh [s2435280@student.ssh.inf.ed.ac.uk](mailto:s2435280@student.ssh.inf.ed.ac.uk) // to get onto uni network

pQnyQ19%p5zW35sD

* ssh [s2435280@mlp.inf.ed.ac.uk](mailto:s2435280@mlp.inf.ed.ac.uk) // to get onto cluster
* move files
  + git clone https://ghp\_r4FQbO6UUD3TCZ3FwrkITg4xdVVRqO0MLeYM @github.com/isebire/summarisation.git
* conda activate edidisscluster
* huggingface-cli login --token hf\_iLYhdGNZTTyzyDchhcwIQDzzcnBPYLYrDT
* sbatch (from the head node – does the allocation)
  + sbatch --time=08:00:00 --mem=14000 --cpus-per-task=4 --gres=gpu:1 test.sh

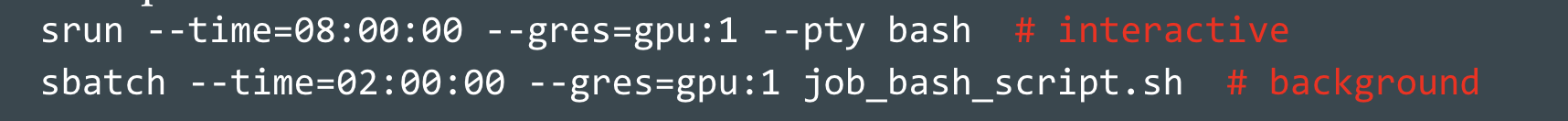
To run the helpful scripts, use ./COMMAND from inside the folder

**Helpful**

* **to remove a directory = rm -rf <name>**

Cluster tips:

* create the script to runA picture containing text, font, screenshot

  Description automatically generated
* run job using slurm
  + use sbatch
* best to copy your code and input data from the distributed filesystem to scratch space (local disk space) at the start of a job, then copy output data back from scratch space to the distributed filesystem at the end of the job.
  + Scratch space usually named /disk/scratch but some nodes have /disk/scratch1 or /disk/scratch2 or /disk/scratch\_big or /disk/scratch\_fast
  + It should also then delete all of your files from scratch space when you are finished using it, so that other people can use it after you. If you are running multiple jobs on the same node or are likely to then it's acceptable to leave the data on between jobs rather than copying it across every time.
* zip everything (code and data) → copy to distributed filesystem using rsync -u → copy to scratch space → unzip → run → zip outputs → copy to distributed filesystem
* set **resource limits** for your jobs. The **--mem** and **--time** options to **sbatch** do this. For details type **man sbatch**. Set the maximum memory to at least 1000MB less than the node's memory divided by the number of GPUs in the node, and set the maximum time to a few hours
* have jobs write output regularly in case of failiure