

Batch jobs

Typically, a job is created via a submission script (shell script) where the first line of the submission file has to be the hashbang `#!/bin/bash` and next lines must be the **SBATCH directives**.

My first job with slurm

The following example is a simple script to help you become familiar with slurm.

Simply make the request for resources (partitions, number of nodes, memory, maximum execution time, working directories, output files, ...), start with an execution of basic linux commands (steps) and wait 120 seconds before to finish the job.

sbatch_script_example.sh

```
#!/bin/bash

#SBATCH --job=test_one #      # Job name
#SBATCH --partition=batch #  # Partition
#SBATCH --nodes=1 #        # Number of nodes
#SBATCH --mem=10gb #       # Job memory request per node (28GB,60GB,200GB)
#SBATCH --tasks-per-node=1 # # Number of task per node
#SBATCH --time=02:00 #     # Time limit
#SBATCH --network=IB #    # Use of Infiniband network (40Gbps)
#SBATCH --output=file_%j.log # # Log file
#SBATCH --error=file_%j.err # # Log file error
#SBATCH --chdir=. #       # Working directory
#SBATCH --mail-user=email # # Where to send email
#SBATCH --mail-type=END,FAIL # # Mail events
#####

source /etc/profile.d/profile.modules.sh

# A COMMENT
echo "Begin my script"
pwd
hostname
date
sleep 120
echo "End my script"
```

You can write this script with *directives short options*:

sbatch_script_example_2.sh

```
#!/bin/bash

#SBATCH -J mi_primer_test # # Job name
#SBATCH -p batch # Partition
#SBATCH -N 1 # Number of nodes
#SBATCH --mem=10gb # Job memory request per node (28GB,60GB,200GB)
#SBATCH --tasks-per-node=1 # Number of task per node
#SBATCH -t 02:00 # Time limit

#SBATCH -o file_%j.log # Log file
#SBATCH --error=file_%j.err # Log file error
#SBATCH -D . # Working directory

#SBATCH --mail-user=emil # Where to send email
#SBATCH --mail-type=END,FAIL # Mail events
#####

source /etc/profile.d/profile.modules.sh

# A COMMENT
echo "Begin my script"
pwd
hostname
date
sleep 120
echo "End my script"
```

Save the file under an appropriate name in a working directory created for it, for example `my_first_test.sh` or `my_first_test.sbatch`.

Info

There are certain options that have default values already set, such as the partition, where the default partition is the `batch` partition.

You can see all options for [sbatch directives](#) in slurm documentation web or running the following command:

```
man sbatch
sbatch --help
sbatch --usage
```

How to running a job

To running a job you only type the following command:

```
SBATCH mi_first_test.sh
or
SBATCH mi_first_test.sbatch
```

My second job with slurm

In this second example **we are going to load an application (module)** to be able to use it. Specifically a python module and we will execute a small python script.

hello_world.py

```
from time import sleep

print ("Hello World")
sleep(120)
print ("Bye world")
```

It is recommended to create a virtual environment to run python (virtualenv, venv, pyenv, conda environment or pipenv). For example, to use `venv`, in the working directory run:

```
module load GCCcore/11.2.0 Python/3.8.6
python3 -m venv /path/workdir/venv
pip install ....
```

- Script for a job:

mi_python_test.sh

```
#!/bin/bash

#SBATCH --job=mi_python_test      # Job name
#SBATCH --nodes=1                # Number of nodes

#SBATCH --time=02:00             # Maximum time limit

#SBATCH --output=file_%j.log      # Standard output log
#SBATCH --error=file_%j.err       # Error output log
#SBATCH --chdir=.                # Working directory

#SBATCH --mail-user=email         # Where to send email
#SBATCH --mail-type=END,FAIL      # Email events
#####

source /etc/profile.d/profile.modules.sh

# before loading the module we see if it has python3 executable
echo "python3 before:"; python3 --version

# Load modules
module purge
```

```
module load GCCcore/11.2.0 Python/3.8.6

echo "python3 despues"; python3 --version
which python3

echo "START STEP1"
# Activating python environment
source /path/workdir/venv/bin/activate
# Launch python script
python3 hello_world.py

echo "END MY SCRIPT"
```

Remember that your working directory is `.`, which is the same directory where y is running script. If `hello_world.py` is in another directory, you must specify the full path.

Another option is to use the **slurm environment variable** `SLURM_SUBMIT_DIR`. To running this job type:

```
sbatch my_second_test.sh
or
sbatch my_second_test.sbatch
```