

Ejecutar un trabajo en la cola

Normalmente, un trabajo es creado mediante un script de lanzamiento (shell script) donde las primeras líneas del archivo deben contener las **directivas SBATCH** así como la línea de comienzo `#!/bin/bash` que indica el interprete de shell a usar.

Mi primer trabajo con slurm

En el siguiente ejemplo es un sencillo script que le ayude a familiarizarse con ellos.

Simplemente realiza la solicitud de recursos (particiones, nº de nodos, memoria, tiempo máximo de ejecución, directorios de trabajo, ficheros de salida,...), comienza con una ejecución de comandos linux básicos (steps) y espera 120 segundos antes de finalizar el trabajo.

SBATCH_script_example.sh

```
#!/bin/bash

#SBATCH --job=mi_primer_test      # Nombre del trabajo
#SBATCH --partition=batch         # Partición (express/sbatch/long/fatnodes)
#SBATCH --nodes=1                # Nº de nodos
#SBATCH --mem=30000M             # Memoria solicitada por nodo (30000M or 60000M)
#SBATCH --tasks-per-node=1       # Nº tareas por nodo
#SBATCH --constrains=sandy       # sandy, ilk (icelake)... architecture
#SBATCH --time=02:00            # Límite de tiempo

#SBATCH --output=file_%j.log     # Log de salida estandar
#SBATCH --error=file_%j.err      # Log de salida errores
#SBATCH -D .                     # Directorio de trabajo

#SBATCH --mail-user=EMAIL        # Donde será enviado el mail
#SBATCH --mail-type=END,FAIL     # Eventos email
#####

# UN COMENTARIO
echo "Comienza mi script"
pwd
hostname
date
sleep 120
echo "Finaliza mi script"
```

El mismo script se puede escribir con la *opciones corta*:

SBATCH_SCRIPT_EXAMPLE_2.SH

```
#!/bin/bash

#SBATCH -J mi_primer_test      # Nombre del trabajo
#SBATCH -p batch              # Partición (express/sbatch/long/fatnodes)
#SBATCH -N 1                  # N° de nodos
#SBATCH --mem=30000M          # Memoria solicitada por nodo (30000M or 60000M)
#SBATCH --tasks-per-node=1    # N° tareas por nodo
#SBATCH --constrains=sandy    # sandy, ilk (icelake)... arquitectura
#SBATCH -t 02:00              # Límite de tiempo

#SBATCH -o file%j.log         # Log de salida
#SBATCH -e file_%j.err       # Log de salida errores
#SBATCH -D .                  # Directorio de trabajo

#SBATCH --mail-user=EMAIL     # Donde será enviado el mail
#SBATCH --mail-type=END,FAIL  # Eventos email
#####

# UN COMENTARIO
echo "Comienza mi script"
pwd
hostname
date
sleep 120
echo "Finaliza mi script"
```

Guarde el fichero creado anteriormente con un nombre apropiado en un directorio de trabajo creado para ello, por ejemplo `mi_primer_test.sh` o `mi_primer_test.sbatch`.

Info

Hay determinadas opciones que tienen valores ya establecidos por defecto, como es la partición, donde la partición por defecto es la `batch`.

Para ver todas las opciones disponibles sobre las **directivas sbatch** puede visitar la documentación oficial de slurm o ejecutando los siguientes comandos:

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

Como lanzar un job con sbatch

Para lanzar el trabajo solo tiene que ejecutar el siguiente comando.

```
sbatch mi_primer_test.sh
or
sbatch mi_primer_test.sbatch
```

Mi segundo trabajo con slurm

En este segundo ejemplo simplemente **cargaremos una aplicación (module)** para poder usarla. Concretamente un módulo de python y ejecutaremos un pequeño script de python.

hello_world.py

```
from time import sleep

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

Es recomendable crear un entorno virtual para ejecutar python (virtualenv, venv, pyenv, conda environment, pipenv). Por ejemplo para usar `venv`, en el directorio de trabajo ejecutar:

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

- Crear el script de lanzamiento:

mi_python_test.sh

```
#!/bin/bash

#SBATCH --job=mi_python_test      # Nombre del trabajo
#SBATCH --nodes=1                 # N° de nodos
#SBATCH --constraints=sandy       # sandy, ilk (icelake)... architecture
#SBATCH --time=02:00             # Límite de tiempo

#SBATCH --output=file_%j.log      # Log de salida estandar
#SBATCH --error=file_%j.err       # Log de salida errores
#SBATCH --chdir=.                 # Directorio de trabajo

#SBATCH --mail-user=email         # Donde será enviado el mail
#SBATCH --mail-type=END,FAIL      # Eventos email
#####

# antes de cargar el módulo vemos si ha ejecutable de python3
echo "python3 antes: "; python3 --version

# Carga modulos
```

```
module purge
module load GCCcore/11.2.0 Python/3.8.6

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

echo "COMIENZA STEP1"
# Activo el entorno de python
source /path/workdir/venv/bin/activate

# Lanzamos el script de python
python3 hello_world.py

echo "FINALIZA MI SCRIPT"
```

Recuerde que su directorio de trabajo es `.`, que es el mismo directorio donde está script de ejecución y. Si `hello_world.py` está en otro directorio, debe especificar la ruta completa.

Otra opción es usar la **variable de entorno de slurm*** `SLURM_SUBMIT_DIR`. Para lanzar el trabajo:

```
sbatch mi_python_test.sh
or
sbatch mi_python_test.sbatch
```