How-to create a simple SLURM job

Make sure you already have an Apptainer container ready to use at this stage

You should have your container ready in the Apptainer file format : for examples here, application.sif

To run your job via SLURM, it is best to create a **sbatch script** :

(Note: for now we don't have Modules or LMod installed, we might add it later : hence the module commands commented)

Example for a serial job

```
#!/bin/bash
#SBATCH --job-name=application   # create a short name for your job
#SBATCH --nodes=1                # node count
#SBATCH --ntasks=1               # total number of tasks across all nodes
#SBATCH --cpus-per-task=1        # cpu-cores per task (>1 if multi-threaded
tasks)
#SBATCH --mem=4G                 # total memory per node (4 GB per cpu-core
is default)
#SBATCH --time=00:05:00          # total run time limit (HH:MM:SS)
#SBATCH --mail-type=begin        # send email when job begins
#SBATCH --mail-type=end          # send email when job ends
#SBATCH --mail-user=email@domain.org
#module purge
apptainer run application.sif <arg-1> <arg-2> ... <arg-N>
```

Example for a parallel MPI code

```
#!/bin/bash
#SBATCH --job-name=solar         # create a short name for your job
#SBATCH --nodes=1                # node count
#SBATCH --ntasks=4               # total number of tasks across all nodes
#SBATCH --cpus-per-task=1        # cpu-cores per task (>1 if multi-threaded
tasks)
#SBATCH --mem-per-cpu=4G         # memory per cpu-core (4G per cpu-core is
default)
#SBATCH --time=00:05:00          # total run time limit (HH:MM:SS)
#SBATCH --mail-type=begin        # send email when job begins
#SBATCH --mail-type=end          # send email when job ends
#SBATCH --mail-user=email@domain.org
#module purge
#module load openmpi/gcc/4.1.2
srun apptainer exec solar.sif /opt/ray-kit/bin/solar inputs.dat
```

Example using GPUs

```
#!/bin/bash
#SBATCH --job-name=tensorflow    # create a short name for your job
#SBATCH --nodes=1                # node count
#SBATCH --ntasks=1               # total number of tasks across all nodes
#SBATCH --cpus-per-task=4        # cpu-cores per task (>1 if multi-threaded tasks)
#SBATCH --mem-per-cpu=4G         # memory per cpu-core (4G per cpu-core is default)
#SBATCH --time=00:05:00          # total run time limit (HH:MM:SS)
#SBATCH --gres=gpu:1             # number of gpus per node
#SBATCH --mail-type=begin        # send email when job begins
#SBATCH --mail-type=end          # send email when job ends
#SBATCH --mail-user=email@domain.org
#module purge
apptainer exec --nv ./tensorflow.sif python3 tensor.py
```

Note the **-nv** flag that allows you to use the GPU from your container without being root.

Example using GPU sharding

Sharding is a generic way of SLURM to use fragments of a GPU for a job, leaving room for other researchers, or running several jobs needing each one a fragment of GPU.

```
#!/bin/bash
#SBATCH --partition=Dance        # Partition to run the job on
#SBATCH --job-name=tensorflow    # create a short name for your job
#SBATCH --nodes=1                # node count
#SBATCH --ntasks=1               # total number of tasks across all nodes
#SBATCH --cpus-per-task=4        # cpu-cores per task (>1 if multi-threaded tasks)
#SBATCH --mem-per-cpu=4G         # memory per cpu-core (4G per cpu-core is default)
#SBATCH --time=02:00:00          # total run time limit (HH:MM:SS)
#SBATCH --gres=shard:24          # number of gpu shards to use
#SBATCH --mail-type=begin        # send email when job begins
#SBATCH --mail-type=end          # send email when job ends
#SBATCH --mail-user=email@domain.org
#module purge
apptainer exec --nv ./tensorflow.sif python3 tensor.py
```

Note the **-nv** flag that allows you to use the GPU from your container without being root. Here, the partition has been requested to be specifically run on the Disco node, which has max 80 shards. On Chacha the limit is 96 shards.

Example using an interactive shell

NOTE : Debugging your application directly on the ISC compute is to be avoided

To debug your application on a test Slurm + Apptainer infrastructure, you can use srun with the –pty argument to run your container :

```
# For a simple compute container :
srun --cpus-per-task=12 --time=3:00:00 --mem=24G --pty apptainer shell
/home/user.name/example_apptainer.sif

# Or a container using GPUs :
srun -G 1 --cpus-per-task=12 --time=3:00:00 --mem=24G --pty apptainer shell
--nv /home/user.name/example_apptainer.sif
```

Example requesting a specific node

NOTE : This is to be avoided : SLURM already chooses the best node matching your requirements if you correctly wrote them in your sbatch : you might just wait in the job queue instead of running directly on another node

If for some specific reasons you want to request a specific node, you can add this parameter to your sbatch :

```
#SBATCH --nodelist=disco                # nodes to use
```

Execute your batch file

Then you can run your sbatch script :

```
sbatch ./application_sbatch.sh
```

From:
https://wiki.isc-vs.ch/ - **The ISC wiki**

Permanent link:
https://wiki.isc-vs.ch/doku.php?id=infra:howto:slurm_sample&rev=1746781497

Last update: **2025/05/09 09:04**