

Writing Your Job Script

The job script is the shell code your job runs on the compute node. Paste it into the **Script / Command** field on the submission form — commands only, nothing else.

What PowerPortal adds automatically

The submit form shows two read-only lines above the editor:

```
#!/bin/bash
source ~/.bashrc
```

These are prepended to every job before submission. You do not need to type them. PowerPortal also sources the module system on the backend, so `module load` works inside your job without any extra setup.

Do not use #SBATCH directives in the script

All resource settings (partition, CPUs, memory, time limit, etc.) are set via the form fields — not via `#SBATCH` lines in the script. If you paste a script that contains `#SBATCH` directives, PowerPortal will strip them automatically and show a warning. Use the form fields instead.

A minimal example

```
module load python/3.11
python my_analysis.py --input data.csv --output results/
```

A more complete example

```
module load openmpi/openmpi-5.0.7-rocky9-slurm
cd $HOME/myproject
python preprocess.py

mpirun -n 16 ./simulate --config config.yaml
```

Working directory

Slurm sets the working directory to the value you enter in the **Working Directory** field. Use absolute paths or `cd` explicitly if your input files are in a different location.

Output and error files

By default, stdout and stderr are written to `slurm-<jobid>.out` in the working directory. Set custom paths in the **Output file** and **Error file** fields. Use `%j` for job ID and `%a` for array task ID in file names.

Tips

- Add `set -e` at the top to stop the script immediately if any command fails.
- Print key variables at the start (`echo "Running on $(hostname)"`) to make debugging easier.
- Test your script interactively with `salloc` before submitting a long job.

Created 2026-06-25 08:31:34 UTC by levk
Updated 2026-07-08 15:36:38 UTC by levk