GPUs, Constraints & Features

Beyond standard CPU/memory resources, the cluster provides special resources you can request via `--gres` and `--constraint`.

GPU Jobs

To request a GPU, use the `gpu-general-pool` partition and add `--gres=gpu:1`:

```
#!/bin/bash
#SBATCH --job-name=gpu_job
#SBATCH --account=my_account
#SBATCH --partition=gpu-general-pool
#SBATCH --qos=my_qos
#SBATCH --time=02:00:00
#SBATCH --ntasks=1
#SBATCH --nodes=1
#SBATCH --cpus-per-task=10
#SBATCH --gres=gpu:1
#SBATCH --mem-per-cpu=4G
#SBATCH --output=gpu_job_%j.out
#SBATCH --error=gpu_job_%j.err

module purge
module load python/python-3.8

# Your GPU commands here
```

Constraints

Constraints let you target nodes with specific hardware. Add `--constraint=<feature>` to your job script:

```
#SBATCH --constraint=localscratch,amd
```

Multiple constraints are comma-separated — the job will only run on nodes matching all of them.

Available Features

Run `features` on the login node to see the current list. Current features:

Feature	Description
`Af3`	Nodes that can run AlphaFold3
`AMD`	Nodes with AMD CPU family
`avx`	Nodes with AVX CPU capabilities
`localscratch`	Nodes with at least 700GB local `/localscratch`
`ib_bh`	Nodes with InfiniBand network #1
`ib_dj`	Nodes with InfiniBand network #2
`Intel`	Nodes with Intel CPU family

Using Local Scratch

If your job uses `/localscratch`, you must clean up after yourself — space is shared across all jobs on that node. Add this to your script:

```
export CACHEDIR=/localscratch/${USER}_${SLURM_JOB_ID}
mkdir -p $CACHEDIR

cleanup() {
  rm -rf -- "$CACHEDIR" || true
}
trap cleanup EXIT INT TERM HUP
```

The `trap` ensures cleanup runs even if the job fails or is cancelled.

In order to add /localscratch feature to your job request, please add this parameter to your job's parameters:

```
#SBATCH --constraint=localscratch
```

Available GRES Types

```
GresTypes=gpu,amd,af3,intel,localscratch
```

Created 2026-06-14 08:16:39 UTC by levk
Updated 2026-08-16 06:04:22 UTC by dvory