Environment

Managing software modules and storage on the cluster.

- Environment Modules
- Storage and Scratch

Environment Modules

Environment Modules let you dynamically load and unload software packages without conflicts between versions. Always load the modules your job needs before running it.

Common Commands

Finding Modules

```
# List all available modules
module avail

# Search for a specific module
module avail gcc

# Get detailed info including dependencies
module spider gcc/gcc-12.1.0
```

Loading and Unloading

```
# Load a module
module load gcc/gcc-12.1.0

# List currently loaded modules
module list

# Unload a specific module
module unload gcc/gcc-12.1.0

# Unload all modules
module purge
```

Inspecting a Module

```
# See what environment variables a module will set
module show gcc/gcc-12.1.0
```

In Job Scripts

Load modules inside your job script, after the `#SBATCH` directives:

```
#!/bin/bash
#SBATCH --job-name=my_job
#SBATCH ...

module purge
module load gcc/gcc-12.1.0

./my_program
```

Starting with `module purge` ensures your job is not affected by any modules loaded in your shell session.

Storage and Scratch

Home Directory

Every user has a personal home directory at `/home/your_username`. This is your default working directory when you log in.

- Backed up via Legato — see TAU backup info
- For purchasing additional storage, see storage pricing
- NetApp storage includes snapshots for file recovery

Scratch Partitions

Scratch partitions are shared, high-speed temporary storage available across the cluster:

- `/scratch100`
- `/scratch200`
- `/scratch300`

Use scratch for intermediate files during a job run — not for long-term storage.

Local Scratch

Some compute nodes and workstations have a local `/localscratch` partition. This is node-local storage — faster than shared scratch but only accessible from that specific node.

If your job uses `/localscratch`, you must clean up after yourself. Add this to your job script:

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

cleanup() {
  rm -rf -- "$CACHEDIR" || true
```

```
}
trap cleanup EXIT INT TERM HUP
```

Important

- **Scratch is not backed up** — do not store anything you cannot afford to lose
- Clean up scratch files after your job completes
- Do not use scratch as a permanent storage location