A6000 Research Server Documentation

April 2025

Overview

This server is intended for deep learning and research workloads by multiple students and researchers. The system has been upgraded with high-performance and large-capacity storage to support parallel experimentation, model development, and data handling.

Hardware Summary

- **Server Name:** A6000 Server
- **Architecture:** x86_64
- **CPU:** 2× Intel(R) Xeon(R) Gold 5418Y, 96 threads (24 cores/socket, 2 threads/core), 3.8GHz max
- **RAM:** 256GB DDR4
- **GPU:** 8× NVIDIA RTX A6000 (48GB each)
- **CUDA Version:** 12.3
- **NVIDIA Driver:** 545.23.08
- **Total Storage:** 40TB usable

Storage Layout

Mount Point	Disk(s)	Capacity	Type	Purpose
/home	2×8TB SSD (RAID0)	15TB	SSD (RAID 0)	User home directories (fast access)
/data	22TB HDD	20TB	HDD	Large datasets, models, experiments
/fast	1.4TB NVMe	1.4TB	NVMe SSD	Conda envs, pip/hf cache, scratch use
/archive	5×1TB HDD (RAID5)	3.6TB	HDD (RAID 5)	Backup/archive, not in active use yet

Shared Conda & Cache Configuration

Conda

- Conda environments: `/fast/conda/envs`
- Conda package cache: `/fast/conda/pkgs`

This is pre-configured via a `.condarc` file in each user's home directory.

HuggingFace / Torch Cache

- HuggingFace cache: `/fast/hf_cache`
- Torch cache: `/fast/hf_cache`

Global environment variables:

```
export CONDA_ENVS_PATH=/fast/conda/envs
export CONDA_PKGS_DIRS=/fast/conda/pkgs
export TRANSFORMERS_CACHE=/fast/hf_cache
export HF_HOME=/fast/hf_cache
export TORCH_HOME=/fast/hf_cache
```

Notes for Users

- All users have access to `/home`, `/data`, and `/fast`
- You are free to create, experiment, and install things as needed
- Use `/fast` for temporary files, cache, training runs
- **Do not** store large datasets or models in `/home`. Use `/data`
- `/archive` is reserved for backups (future use)

System Admin Notes

- RAID0 setup via `mdadm` for `/home`
- RAID5 setup via `mdadm` for `/archive`
- `/etc/fstab` and `mdadm.conf` updated for persistence
- All user caches cleaned (April 2025) and redirected to `/fast`

Optional Future Enhancements

- Auto-cleanup for `/fast/scratch`
- Quotas for `/home`
- Cron-based backups to `/archive`