ilifu Online Training

Session 1: Introduction to Slurm

24 February 2026

Dane Kennedy · ilifu Bioinformatics Support
University of Cape Town

View on github.io

printable version

Job Scheduling & Cluster Management Tool

An open-source **job scheduling & cluster management tool**. You submit work from the login node; Slurm allocates compute resources and runs your jobs when they're available.

- Login Node
  - Accessed via ssh (`$ ssh <username>@slurm.ilifu.ac.za`)
  - Submit jobs and basic file management
- Scheduler and Accounting Database
  - Manage jobs, Partitions and Nodes
  - Accounting
- Compute Nodes
  - Where your analysis / simulation runs (inside of a slurm job)
  - Software available via singularity containers or modules

Specific node / partition use cases

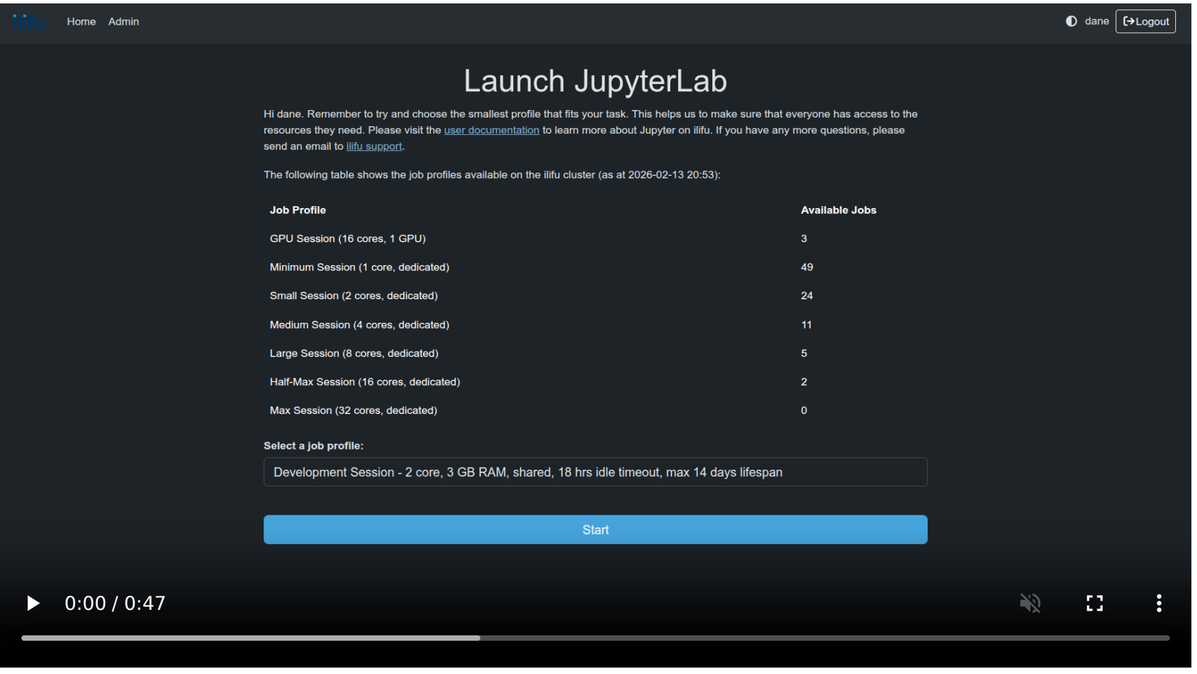
Node / Partition	Use Case	Resources
`slurm.ilifu.ac.za`	SLURM & bash commands (`cd` , `mkdir` , `sbatch`)	—
`compute-001` / Devel	Development, debugging, testing, interactive jobs	1 × (32 cores, ~232 GiB)
`jupyter-0[01-14]` / Jupyter	JupyterLab — interactive development/analysis	14 × (32 cores, ~232 GiB)
`compute-0[02-86]` / Main	Stable, heavy computation	85 × (32 cores, ~232 GiB)
`highmem-00[1-8]` / HighMem	High memory jobs	6 × (32 cores, 500 GiB) + 1 × (32 cores, 1 TiB) + 1 × (96 cores, 1.5 TiB)
`gpu-00[1-7]` / GPU	GPU-accelerated workloads	7 × (32 cores, ~232 GiB + NVIDIA GPUs)

Available Resources

Interacting with the Slurm Cluster

🌐 Web Access

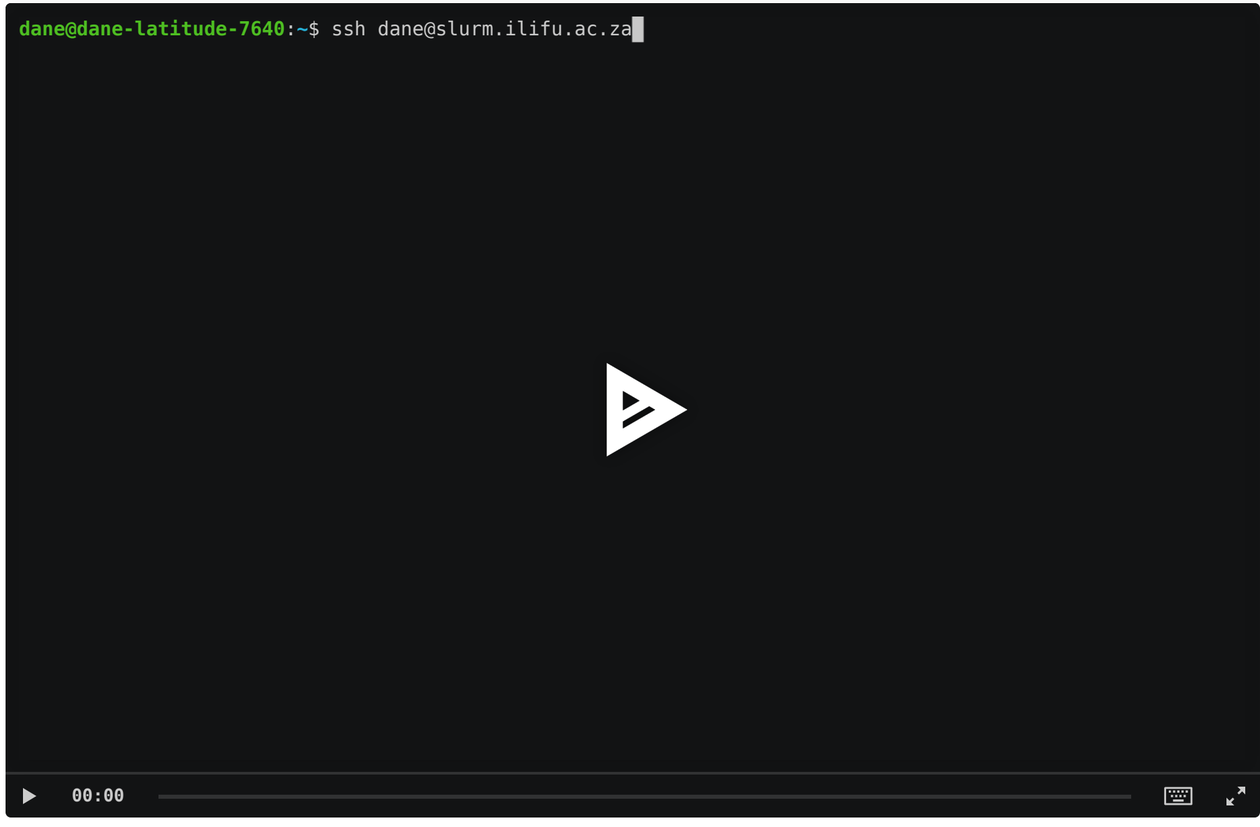
https://jupyter.ilifu.ac.za



> 💡 **Remember to end your session:** File → Hub Control Panel → Stop My Server.

💻 CLI Access

```
ssh <username>@slurm.ilifu.ac.za
```



CLI user commands

```
ssh <username>@slurm.ilifu.ac.za  # connect to login node
sinfo                             # show partition information
squeue                            # show all job information
squeue -u $USER                   # show only your jobs
sbatch your_job_script.sh         # submit a job
scancel                           # cancel a job
man sbatch                        # show the manual page for a command
sbatch --help                     # show the help page for a command
```

Slurm Quick Start User Guide

Example bash script

```
#!/bin/bash

module add python/3.11.2
python hello_world.py
```

Default Job Parameters

```
#SBATCH --time 0-03:00:00  # 0 days + 3 hours
#SBATCH --mem 3G           # 3 GiB
#SBATCH --ntasks 1         # one task
#SBATCH --nodes 1          # one node
#SBATCH --partition Main
#SBATCH --account <your default>
```

sbatch parameters

Example Slurm Job Script

R_container.sbatch

```
#!/bin/bash
#SBATCH --job-name=tutorial2_R_container
#SBATCH --time=00-00:01:00
#SBATCH --mem=4G
#SBATCH --partition=Main
#SBATCH --output=R_container-%j.stdout
#SBATCH --error=R_container-%j.stderr
#SBATCH --mail-user=YOUR_EMAIL_ADDRESS
#SBATCH --mail-type=BEGIN,END,FAIL,TIME_LIMIT_80
#SBATCH --account=ACCOUNTING_GROUP

singularity exec /software/common/containers/RStudio2023.06.1-524-R4.3.1.sif Rscript hello_world.R
```

Submit the job:

```
sbatch R_container.sbatch
```

Slurm sbatch documentation

Defaults and maximums per partition

Partition	Node names	Default CPUs	Max CPUs	Default Memory (GiB)	Max Memory (GiB)	Default wall-time	Max wall-time
Main	compute-[002-021]	1	32	3	232	3 hours	14 days
Main	compute-[101-105]	1	48	3	232	3 hours	14 days
Main	compute-[201-260]	1	32	3	251	3 hours	14 days
HighMem	highmem-[001-002]	1	32	15	503	3 hours	14 days
HighMem	highmem-003	1	96	15	1508	3 hours	14 days
HighMem	highmem-[004-007]	1	32	15	503	3 hours	14 days
HighMem	highmem-008	1	32	15	1007	3 hours	14 days
GPU	gpu-[001-004]	1	32	7	232	3 hours	14 days
GPU	gpu-005	1	24	7	232	3 hours	14 days
GPU	gpu-006	1	48	7	354	3 hours	14 days
GPU	gpu-007	1	48	7	354	3 hours	14 days
GPU	highmem-008	1	32	7	1007	3 hours	14 days
Devel	compute-001	1	32	-	-	3 hours	5 days

Maximum Allocation

Demo

minimal.sbatch

maximal_python.sbatch

hello_world.py

maximal_R.sbatch

hello_world.R

Best practices

> Remember that the ilifu cluster is a shared resource with limited resources. It is important to be mindful of this when requesting resources.

✅ Do

- Use `sbatch` for production work
- Identify resource requirements first:
  - Number of nodes
  - Cores per node
  - Ram per core/node
  - Expected runtime
- Clean up temp files (e.g., `/scratch3`)
- Use Singularity containers for software
- Use `transfer.ilifu.ac.za` for data transfers

❌ Don't

- Run heavy processes on the login node
- Place large files in `/users` (home dir)
- Use `scp` / `rsync` on login node
- Request way more resources than needed

Best Practices

Thank you for your time!

We hope this presentation was helpful.

Contact Support

support@ilifu.ac.za

Documentation

docs.ilifu.ac.za