# SLURM HANDS-ON EXERCISE

In this tutorial, we're going to walk through a complete hands-on SLURM workflow, starting with submitting a job, hitting an error, and then debugging it using the right commands.

---

## Create a Simple Batch Job

Open a terminal and create a file:

```
$ nano myjob.sh
```

```
#!/bin/bash
#SBATCH --job-name=myjob
#SBATCH --output=myjob_%j.out
#SBATCH --ntasks=1
#SBATCH --nodes=1
#SBATCH --time=00:06:00
#SBATCH --partition=testing
echo "Running job on $(hostname)"
sleep 120
echo "Done at $(date)"
```

## Submit the script

```
$ sbatch myjob.sh
```

This will give an invalid partition error:

```
sbatch: error: invalid partition specified: testing
sbatch: error: Batch job submission failed: Invalid partition name specified
```

## Use sinfo to Check Cluster & Partitions

```
$ sinfo
```

Or

```
$ sinfo -s # Provides a summary report
```

Look for columns like:

- PARTITION — list of valid queues
- AVAIL, TIMELIMIT, NODES, STATE

Based on the sinfo output, you will see that the job submission failed because the requested partition doesn't exist. Change the partition name in your batch script:

```
$ nano myjob.sh
```

```
#SBATCH --partition=atesting
```

Save and then submit the job again

```
$ sbatch myjob.sh
```

You should see:

```
Submitted batch job <JOB ID HERE>
```

Example:

```
Submitted batch job 14100630
```

The number listed is the Job ID for the batch script you submitted. Think of it like a tracking number which you can use to check on your job.

## Checking the Job Queue with squeue

```
$ squeue -u $USER
```

The STATUS column will initial say PD (pending), but once your job has started it will change to R (running). After your job has finished running it will disappear from squeue's list.

Your job might sit pending for a short time even in debug, that's normal.

Example:

```
[mokh8410@login-ci5 mokh8410]$ squeue -u $USER
JOBID PARTITION     NAME   USER    ST  TIME  NODES NODELIST(REASON)
14100630  atesting myjob   mokh8410  R  0:19  1     c3cpu-a2-u3-1
```

## Checking the Job's Log

When slurm starts a job, it will create an associated log file. This log file captures the job's standard output (e.g. print statements, error messages, etc.) and is a handy way to see what happened while the job was running.

The log file's name is set by the "output" directive:

```
#SBATCH --output=myjob_%j.out
```

To check the output file for your job:

```
$ cat myjob_<JOB_ID_HERE>.out
```

Example:

```
$ cat myjob_14100630.out
---
Running job on c3cpu-a2-u3-1.rc.int.colorado.edu
Done at Mon May 19 14:08:21 MDT 2025
---
```

## Job Efficiency Check

You just submitted and completed a Slurm job! It ran on a compute node, and your output was captured in a log. Once the job has completed, you will be able to check on its efficiency – i.e. how much of the requested resources were used. To do this you will need to run the "seff" command:

```
$ seff <JOB_ID_HERE>
```

Example:

```
$ seff 14100630
```

Job ID: 14100630

Cluster: alpine

User/Group: mokh8410/mokh8410pgrp

State: COMPLETED (exit code 0)

Cores: 1

CPU Utilized: 00:00:00

CPU Efficiency: 0.00% of 00:00:42 core-walltime

Job Wall-clock time: 00:00:42

Memory Utilized: 396.00 KB

Memory Efficiency: 0.01% of 3.75 GB

## Review a Submitted Job's Information

As you develop and test your batch scripts, they will inevitably change over time which can make it tricky to remember which version of your batch script a given job used. Luckily, slurm provides an easy way to lookup a job's batch script:

```
$ sacct -j <JOB_ID> -B
```

In addition to viewing the job's batch script, you may want to review a jobs:

Command Line Arguments:

```
$ sacct -j <JOB_ID> --format=SubmitLine -p
```

Working Directory:

```
$ sacct -j <JOB_ID> --format=workdir -p
```

A full list of job fields that can be accessed through sacct are listed in Slurm's.

So to sum it up, we just walked through the full job lifecycle in SLURM:

- We wrote a basic batch script
- Submitted the job and saw it fail
- Used sinfo to fix the partition

- Monitored the job with squeue
- Checked output logs
- Measured efficiency with seff
- And reviewed job details with sacct

These are foundational skills that'll help you troubleshoot and optimize your SLURM workflows no matter what kind of research you're running.