Submitting an Interactive Job

Submitting an interactive job is done with the command `srun`.

```
$ srun --pty $SHELL
```

or to allocate 4 cores per node:

```
$ srun --nodes=1 --ntasks-per-node=4 --mem-per-cpu=1024 --pty $SHELL
```

Submitting an interactive job is useful if you require extra resources to run some processing by hand. It is also very useful to debug your processing.

And interactive job is scheduled onto a worker node just like a regular job. You can provide options to the interactive job just as you would a regular SLURM job.

Priority for short jobs

To run short jobs for testing and development work, a job can specify a different quality of service (QoS). The `short` QoS increases a jobs priority so it will run as soon as possible.

**SLURM Specification**

```
--qos=short
```

**Limits per user for 'short' QoS**

- 6 hour job run time
- 1 job of 16 CPUs or fewer
- No more than 256 CPUs in use for short jobs from all users

**Using the short QoS**

```
srun --qos=short --nodes=1 --ntasks-per-node=1 --mem-per-cpu=1024 --pty $SHELL
```

The `/home` directories are read-only on the worker nodes. You will need to compile or run your processing in `/work`. 

The `/home` directories are read-only on the worker nodes. You will need to compile or run your processing in `/work`. 

The `/home` directories are read-only on the worker nodes. You will need to compile or run your processing in `/work`. 

The `/home` directories are read-only on the worker nodes. You will need to compile or run your processing in `/work`. 

The `/home` directories are read-only on the worker nodes. You will need to compile or run your processing in `/work`. 

The `/home` directories are read-only on the worker nodes. You will need to compile or run your processing in `/work`. 

The `/home` directories are read-only on the worker nodes. You will need to compile or run your processing in `/work`. 

The `/home` directories are read-only on the worker nodes. You will need to compile or run your processing in `/work`. 

The `/home` directories are read-only on the worker nodes. You will need to compile or run your processing in `/work`. 

The `/home` directories are read-only on the worker nodes. You will need to compile or run your processing in `/work`. 

The `/home` directories are read-only on the worker nodes. You will need to compile or run your processing in `/work`. 

The `/home` directories are read-only on the worker nodes. You will need to compile or run your processing in `/work`. 

The `/home` directories are read-only on the worker nodes. You will need to compile or run your processing in `/work`. 

The `/home` directories are read-only on the worker nodes. You will need to compile or run your processing in `/work`. 

The `/home` directories are read-only on the worker nodes. You will need to compile or run your processing in `/work`. 

The `/home` directories are read-only on the worker nodes. You will need to compile or run your processing in `/work`. 

The `/home` directories are read-only on the worker nodes. You will need to compile or run your processing in `/work`. 

The `/home` directories are read-only on the worker nodes. You will need to compile or run your processing in `/work`. 

The `/home` directories are read-only on the worker nodes. You will need to compile or run your processing in `/work`. 

The `/home` directories are read-only on the worker nodes. You will need to compile or run your processing in `/work`. 

The `/home` directories are read-only on the worker nodes. You will need to compile or run your processing in `/work`. 

The `/home` directories are read-only on the worker nodes. You will need to compile or run your processing in `/work`. 

The `/home` directories are read-only on the worker nodes. You will need to compile or run your processing in `/work`. 

The `/home` directories are read-only on the worker nodes. You will need to compile or run your processing in `/work`.