

# Jupyter Notebook Scheduler

- [Usage](#)
- [Examples](#)
- [Restrictions](#)

## Usage

```
$ jup_sched -h
usage: jup_sched [-h] [-e EMAIL] [-t TIME] [--gpu] [--password]

Jupyter Notebook Scheduler

optional arguments:
  -h, --help            show this help message and exit
  -e EMAIL, --email EMAIL
                        specify an email address for the scheduler
                        communication
  -t TIME, --time TIME  specify the running time for the scheduler in the
                        format HH:MM:SS
  --gpu                use this option if you require a GPU to be scheduled
  --password            re-print the password and address for the jupyter
                        notebook WILL IGNORE ALL OTHER ARGUMENTS
```

## Examples

- To get a simple Jupyter Notebook running on one CPU core on a compute node:

```
$ jup_sched
[jupyter_scheduler INFO] no time on arguments, scheduling notebook for 1 HOUR
[jupyter_scheduler INFO] Waiting for scheduler...

=====
Please direct your browser to:
https://nano.ncsa.illinois.edu:10087
Use the password to login:
a25f9815bc783d0e3c7d
=====

Run "jup_sched --password" to view this message again
```

Open a browser on your local computer (not Nano), navigate to the provided address and use the provided password to log in. The job will terminate when the time limit is reached or if you exit via the web interface.

- To get a Jupyter Notebook for two hours:

```
jup_sched -t 02:00:00
```

- To get a Jupyter Notebook with a GPU:

```
jup_sched --gpu
```

## Restrictions

- If you have Anaconda or some other form of user-space Python installation, it will be used instead of the system-wide Python installation. This will allow you to use your packages.
- Only one CPU core and up to one GPU can be allocated for a Jupyter Notebook job.
- Maximum run time is limited to 12 hours.
- This functionality is intended for interactive development and prototyping only, larger computations should be done with regular batch jobs.