Monitoring jobs with qstat

The simplest command for monitoring the state of a job is `qstat`. Run without options, `qstat` will produce a long list of every job queued and running on the system, probably more than you wish to see. To see the state only of your own jobs, use:

```bash
$ qstat -u NetID
```

For example:

```bash
$ qstat -u $USER
```

```
soho.es.its.nyu.edu:
Job ID    Username  Queue   Jobname          SessID  NDS  TSK
Memory Time  S Time
-------------------- -------- -------- ---------------- ------ ----- ---
------ ----- - ----- ------
3593014  ab123    s48      model_scen_1       1584     1  12
2gb  44:0 R  7:06
3593016  ab123    s48      model_scen_2      31262     1  12
2gb  44:0 R  7:05
3593017  ab123    s48      model_scen_3       7443     1  12
2gb  44:0 R  7:05
3593018  ab123    s48      model_scen_4      15454     1  12
2gb  44:0 R  7:05
3601576  ab123    s48      model_scen_5      20458     1  12
4gb  44:0 R  5:31
```

In Unix, the shell sets an environment variable `USER` to your username (at NYU this is your NetID). In the example above this environment variable is used instead of explicitly typing my NetID.

To see the status of a single job - or a list of specific jobs - pass the Job IDs to `qstat`, as in the following example:

```bash
$ qstat 3593014 3593016
```

```
Job id        Name             User            Time Use S Queue
------------- ---------------- --------------- -------- - ----- ------
3593014       model_scen_1     ab123            7:23:47 R s48
3593016       model_scen_1     ab123            7:23:26 R s48
```

Most of the fields in the output are self-explanatory. **The second-last column “S” is the job status.** which can be:

- Q meaning “Queued”
- H meaning “Held” - this may be the result of a manual hold or of a job dependency
- R meaning “Running”
- C meaning “Completed”. After the job finishes, it will remain with “completed” status for a short time before being removed from the batch system.

Other, less common job status flags are described in the manual (`man qsub`).

Note that the output format in this example differs from that of the first example, which shows the time, memory and total number of nodes and tasks requested as well as the elapsed time. To see this extra information add the “-a” switch to `qstat`:
<table>
<thead>
<tr>
<th>Job ID</th>
<th>Username</th>
<th>Queue</th>
<th>Jobname</th>
<th>SessID</th>
<th>NDS</th>
<th>TSK</th>
</tr>
</thead>
<tbody>
<tr>
<td>3593014</td>
<td>ab123</td>
<td>s48</td>
<td>model_scen_1</td>
<td>1584</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>3593016</td>
<td>ab123</td>
<td>s48</td>
<td>model_scen_2</td>
<td>31262</td>
<td>1</td>
<td>12</td>
</tr>
</tbody>
</table>

For detailed information about a specific job, `qstat -f` produces about a page of output detailing the resources requested, resources used, nodes on which the job is running and much more:

```
$ qstat -f job-id
```

Finally, for more options and more detail on output of `qstat`, see the manual page:

```
$ man qstat
```