Jobs are submitted with the `qsub` command:

```
$ qsub options job-script
```

The options tell Torque information about the job, such as what resources will be needed. These can be specified in the job-script as PBS directives, or on the command line as options, or both (in which case the command line options take precedence should the two contradict each other). For each option there is a corresponding PBS directive with the syntax:

```
#PBS option
```

For example, you can specify that a job needs 2 nodes and 8 cores on each node by adding to the script the directive:

```
#!/bin/bash
#PBS -l nodes=2:ppn=8
```

or as a command-line option to `qsub` when you submit the job:

```
$ qsub -l nodes=2:ppn=8 my_script.q
```

**What resources should I request?**

**Requesting exclusive use of nodes**

There are two ways you can ensure you will not share a node with other jobs:

- Use the `"-n"` `qsub` option
- Request nodes with `"-l nodes=num:ppn=all"`
  
  Note that with the second option, you do not know in advance how many cores will be available on the nodes your job is allocated to. More on this below.

Until recently, requesting nodes but not `ppn` gave whole nodes. However this led to users inadvertently requesting full nodes, so now to request full node you use "`ppn=all" instead
PBS directives and qsub options for setting resource limits

Options to request compute resources:

- `-l walltime=walltime`
  Maximum wallclock time the job will need. Default depends on queue, mostly 1 hour. Walltime is specified in seconds or as hh:mm:ss or mm:ss.

- `-l mem=memory`
  Maximum memory per node the job will need. Default depends on queue, normally 2GB for serial jobs and the full node for parallel jobs. Memory should be specified with units, eg 500MB or 8GB. The available memory per node for different nodes of Mercer is described here.

- `-l procs=num`
  Total number of CPUs required. Use this if it does not matter how CPUs are grouped onto nodes - eg, for a purely-MPI job. Don't combine this with `-l nodes=num` or odd behavior will ensue.

- `-l nodes=num:ppn=num`
  Number of nodes and number of processors per node required. Use this if you need processes to be grouped onto nodes - eg, for an MPI/OpenMP hybrid job with 4 MPI processes and 8 OpenMP threads each, use `-l nodes=4:ppn=8`. Don't combine this with `-l procs=num` or odd behavior will ensue. Default is 1 node and 1 processor per node. When using multiple nodes the job script will be executed on the first allocated node.
  Torque will set the environment variables PBS_NUM_NODES to the number of nodes requested, PBS_NUM_PPN to the value of ppn and PBS_S_NP to the total number of processes available to the job.

- `-l nodes=num:ppn=num:ppnum=titan`
  If your job needs GPUs, you should specify the number of GPUs per node in the `-l nodes` option.
  We have nodes with older Tesla GPUs and newer Titan GPUs - if you specifically need the newer GPUs you should add the requirement ":titan" to the gpus specification.

- `-l nodes=num:ppn=all`
  This is an NYU HPC extension: if you need whole nodes but do not mind how many cores, you can request full nodes this way. Specifying `-n` will have the same effect. The environment variable $PSB_PPN will be set in the job to the number of cores on each node the job is running on. Jobs are always allocated to sets of nodes with the same number of cores, so you will not get one node with 12 cores and another with 20.
  Note that this currently only works when used in directives, not on the command line.

- `-n`
  Request exclusive use of nodes. If this is specified, no other jobs will share a node with this job, and if you did not specify a memory limit, no memory limit will be enforced (Note however that if you do not specify a memory limit, you may land on a node with only 24GB of memory).

- `-q queue`
  Submit to a specific queue. If not specified, Torque will choose a queue based on the resources requested.

In almost all cases, it is best not to specify a queue - the system will choose the best queue according to the resources you request.

Resources can be requested with multiple `-l` options, or as a comma-separated list of options. Both of the following examples are correct:

```bash
$ qsub -l walltime=2:00:00 -l mem=4GB myjob.q

$ qsub -l walltime=2:00:00,mem=4GB myjob.q
```

Most nodes on Bowery have 24GB memory. Requesting a large portion of the memory on a node will cause Moab to reserve an entire node for your job even if you only request 1 CPU, since there will be insufficient remaining memory to run other jobs.

The serial queues on NYU HPC clusters are limited to a single node, but allow multiple processors on that node to be used. Therefore, parallel jobs using only one node, such as OpenMP or multithreaded jobs can be submitted to a serial queue.

When using more than one node, the job script is executed only on the first node. To make use of the other nodes you must use MPI or pbsdsh.

The MPI modules on Bowery are not compiled with support for Torque, so the CPU time reported by Torque will include only the CPU time for the first MPI process. For an estimate of the true total CPU time, use the elapsed (wallclock) time multiplied by the number of nodes. This issue has been solved on Mercer.