**Introduction**

The requirement is for creation of a job queue with two parts:

1. Client
2. Worker

The job consists of the following components:

1. A docker image
2. An array of command (cmd) parameters to be passed to the image.
3. A dictionary of CPU and memory requirements

The number of jobs is created by the client and pushed into the job queue. The worker pops the jobs and executes them.

**Job Execution Details**

* There are no restrictions on how long a task can run; Some may finish in weeks while others may run for weeks.
* Assume jobs are idempotent; Each job should be run at least once.
* Each worker should take CPU and memory available as inputs when it's started.
* A worker should simultaneously run as many jobs as possible without overrunning either the CPU or the memory available
* We must be able to run multiple workers with different CPU and memory availability simultaneously.

**Technical objects**

The technical objects for the job queue such as environment variables dictionary, CPU and memory requirements dictionary, Image file details and command line parameter lists have been created using Python code. These objects have been used in the job creation and further processing of the jobs

**Implementation using Python**

The implementation of the object creation, job creation and Parallel processing options has been done using Python. The “multiprocessing” library in Python has been used for the creation and processing of the jobs. The Python code has been uploaded to the GitHub repository.