

# How to use clustering for workers

Let's say you wish to partition your render farm into 2 departments, post and promo. Qube! supports the idea of partitioning your render farm into "clusters". You can divide up your farm between any number of departments, while still allowing the departments to share the machines.

As an example, you could allocate some of your render nodes to the /post cluster, and some to the /promo cluster (Qube! workers - your render nodes - belong to 1 and only 1 cluster). Users would submit their jobs to either the /post cluster or the /promo cluster - this can be configured so that it's transparent to the user.

A worker in the /post cluster will run the lowest-priority job in the /post cluster before it will run the highest-priority job in the /promo cluster.

A worker in the /promo cluster will run the lowest-priority job in the /promo cluster before it will run the highest-priority job in the /post cluster.

You could also add machines to the / cluster (the "root" cluster). Workers in the / cluster will give equal priority to jobs from all departments.

With respect to user permissions, the default in Qube! is that a user may only manipulate (pause, kill, etc) their own jobs. Certain users can be granted the Qube! "admin" privileges, which will allow them the rights to manipulate all users' jobs.

## Clustering and restrictions

Optionally, some or all workers in either the /post or /promo cluster can be configured to ~only~ run jobs in their respective clusters via the [worker\\_restrictions](#) value. You can have some of the /post workers be available to promo department if there are no post jobs, while other /post workers will remain dedicated to the post department.

## More about clustering

- [How clustering affects priority and worker selection for jobs](#)
- [worker\\_cluster](#)
- [How to use qbwrk.conf](#)