# Restarting Commands

Sometimes it's useful to have commands that exited with a non-zero status to restart automatically.

concurrently lets you configure how many times you wish for such a command to restart through the `--restart-tries` flag:

$ concurrently --restart-tries 2 "exit 1"

[0] exit 1 exited with code 1

[0] exit 1 restarted

[0] exit 1 exited with code 1

[0] exit 1 restarted

[0] exit 1 exited with code 1

Sometimes, it might be interesting to have commands wait before restarting.

To do this, simply set `--restart-after` to a the number of milliseconds you'd like to delay restarting.

$ concurrently -p time --restart-tries 1 --restart-after 3000 "exit 1"

[2024-09-01 23:43:55.871] exit 1 exited with code 1

[2024-09-01 23:43:58.874] exit 1 restarted

[2024-09-01 23:43:58.891] exit 1 exited with code 1

If a command is not having success spawning, you might want to instead apply an exponential back-off.

Set `--restart-after exponential` to have commands respawn with a `2^N` seconds delay.

$ concurrently -p time --restart-tries 3 --restart-after exponential "exit 1"

[2024-09-01 23:49:01.124] exit 1 exited with code 1

[2024-09-01 23:49:02.127] exit 1 restarted

[2024-09-01 23:49:02.139] exit 1 exited with code 1

[2024-09-01 23:49:04.141] exit 1 restarted

[2024-09-01 23:49:04.157] exit 1 exited with code 1

[2024-09-01 23:49:08.158] exit 1 restarted

[2024-09-01 23:49:08.174] exit 1 exited with code 1