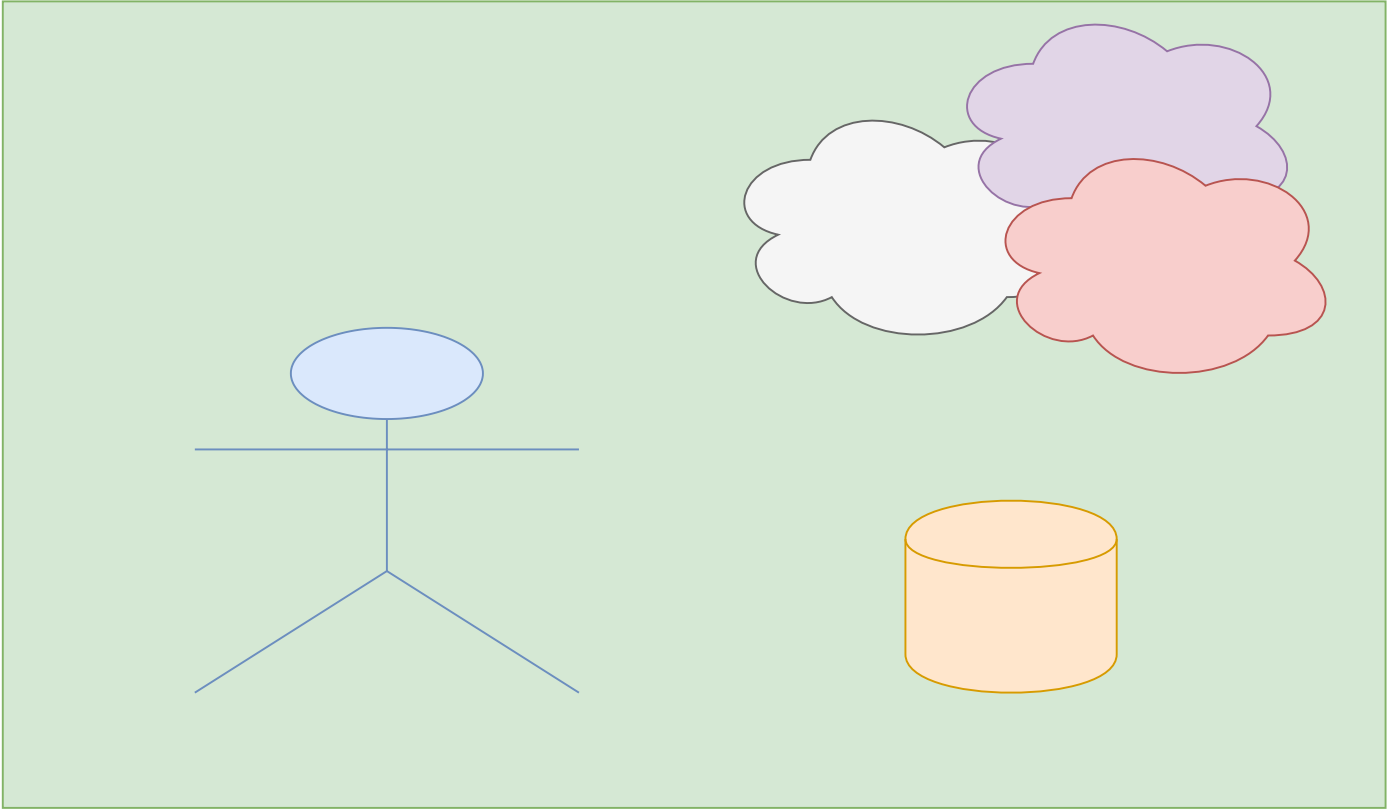
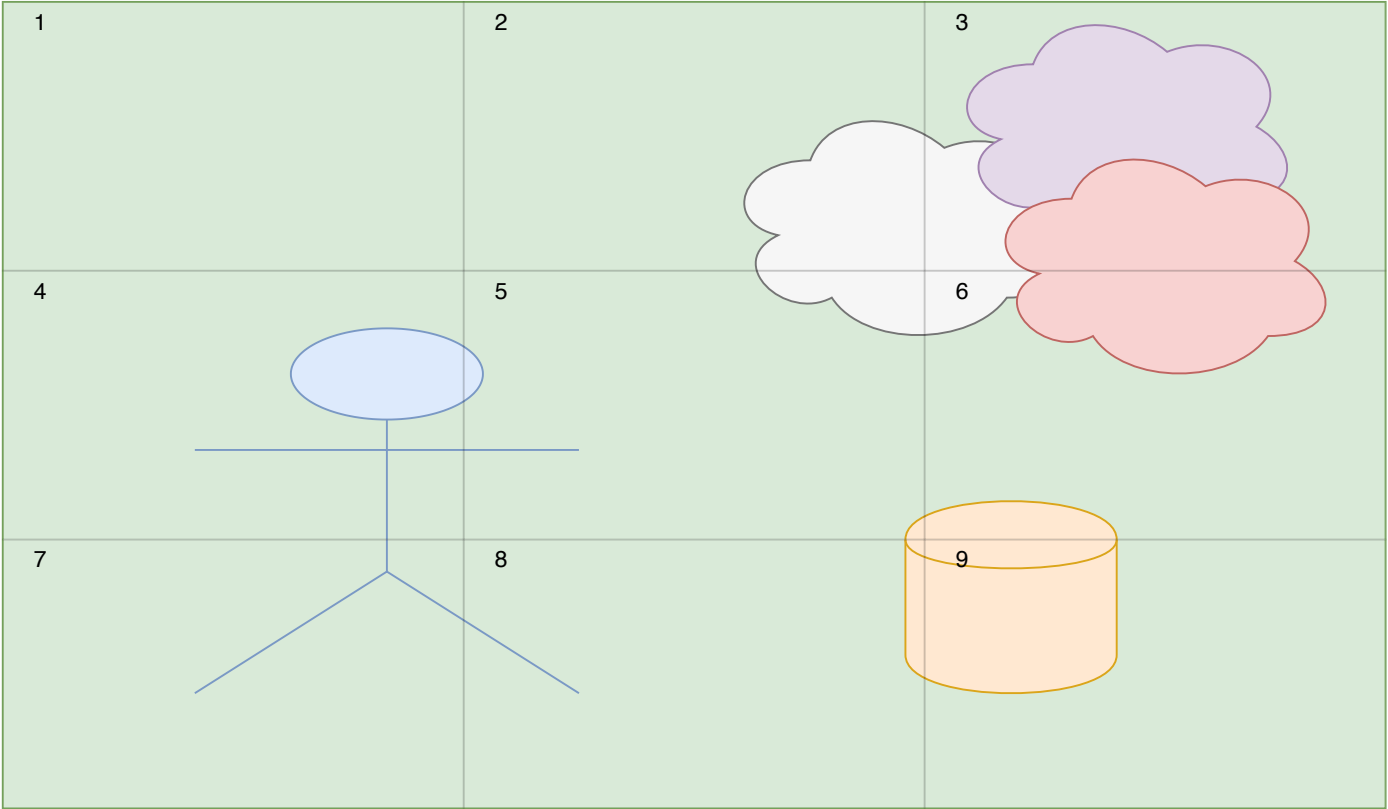


GEOMETRIC DISTRIBUTION

Considering the following picture as the expected output (It should normally looks like to a mandelbrot fractal).



First we need to split the picture in sub rectangles. Each rectangle will be processed by a single goroutine.



The number of sub-rectangle is defined by the method call "runtime.NumCPU" which returns the ideal number of thread you can run considering your current hardware.

The chosen approach is the following:

- Loop over all sub-rectangle
- Launch a goroutine for each and adds it to the waitgroup
- Wait for all goroutine to be processed

This approach has one weakness:

sub-rectangle°1 will be processed very fast as it is empty.  
However sub-rectangle°3 could take longer as it is more complex.

Conclusion:

In a case where only a small portion is very complex, using concurrency won't provide much benefits in term of performance