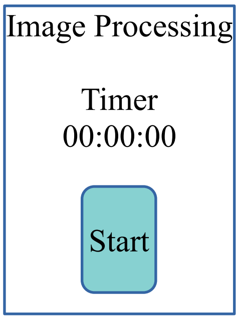
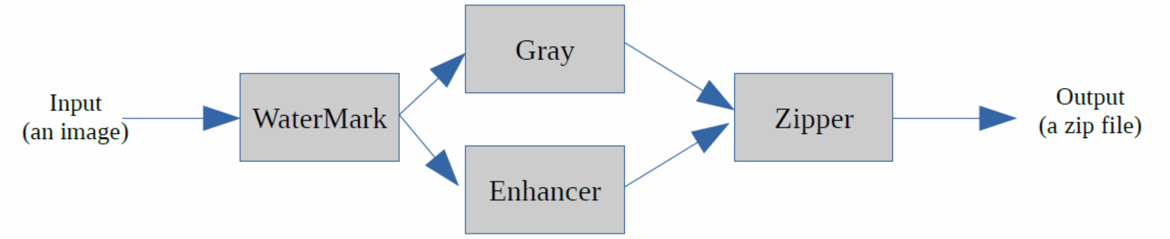
Assignment 6: Implementing parallel workflows in OpenWhisk and Kafka.

Implement the following web page containing a start button and a timer.



When the user clicks the start button, your application should insert all the photos in a photo album to a Kafka topic named topic_1. The photo album is available here. All students should use the same photo album to make it possible to compare different methods.



We consider the same workflow as the assignment 4 where there are 4 functions. The first function (i.e., Watermark) checks Kafka topic_1 for new images. When a new image is arrived in the topic_1, then this function takes it, adds a watermark on it, and submits the wetermarked image to Kafka topic_2.

The other two functions named Enhancer and Grayscale check Kafka topic_2 for new images. When new image is arrived, these functions add their corresponding filter on the image and insert the results in Kafka topic_3.

Zippier function is the last task of this workflow. When the grayscaled and enhanced images are successfully submitted, Zipper should take both images including the grayscaled and enhanced versions of the same image, zip them, and submit them to CouchDB. When all images are successfully uploaded to CouchDB, then the timer should stop for calculating the execution time.

Note that the main purpose of this assignment is to explore how multiple tasks in a fork (i.e., Grayscale and Enhancer) can execute in parallel and also designing an efficient and high performance algorithm capable of reducing the finish time of the application as much as possible.

It is possible to use periodic triggers for parallelism, but it is not an efficient strategy in terms of performance to complete the application. Hence, it is desirable to have an event-based scheduling of functions of the workflow rather than periodic invocations. Nevertheless, we prefer to use event-triggers in Kafka, MessageHub or something like that rather than hardcoding the triggers since on that case you face with synchronization issues, and your code doesn’t easily extend for other workflows!

It is up to you to explore which method can help you develop an efficient algorithm. At the end of this assignment you have to report the best timer values.

The deadline for this assignment is in two weeks (12.00 o’clock, 26 Nov). There are some extra bonus for the best algorithm with the minimum completion time. It is also possible to use IBM MessageHub for those who has credit Cards.