CLOUD COMPUTING

Lab 2 : Project report

Submitted by :

Nassima BEN BAHTANE
Mohammed BENMOUSSA
Mouad RIALI
Amina ZOUARHI

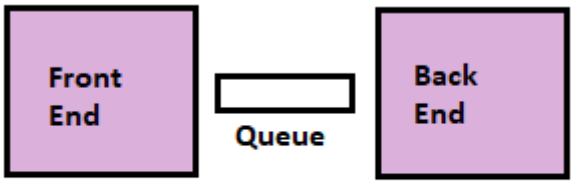
Under the guidance of :

Ms. Charlotte LACLAU

2021 - 2022

Project Architecture :

The project has a classic Web-Queue-Worker architecture. The core components of this architecture are a web front-end that serves client requests, and a worker that performs tasks. The client communicates with the worker through a **message queue**.



Web-Queue-Worker

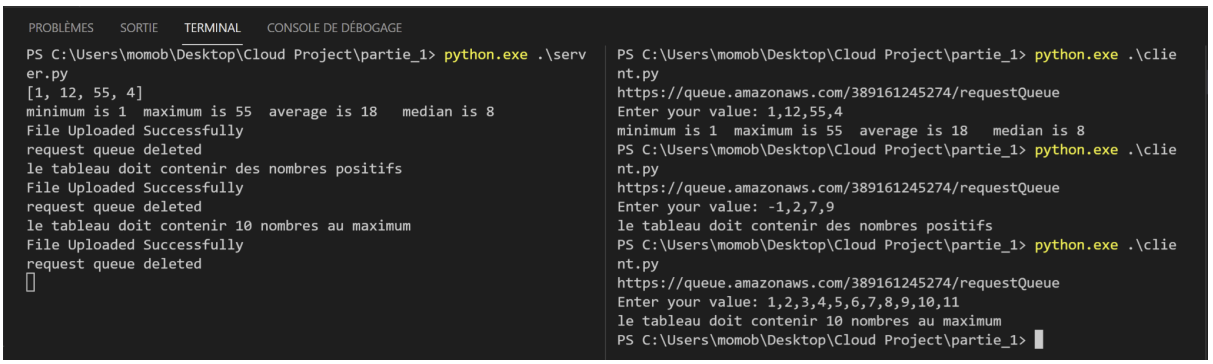
When executed (python.exe .\client.py), the client is asked to enter a list of up-to 10 positive integers. This list is then submitted in a SQS queue for requests.

The worker, which we execute with the command : **python.exe .\sever.py** , is a multi-client server that awaits SQS queues for requests. It reads the integers' list, calculates its minimum, maximum, average and median then submits those results in an SQS queue for the response. A log file containing the results is also submitted to the AWS cloud as a S3 object.

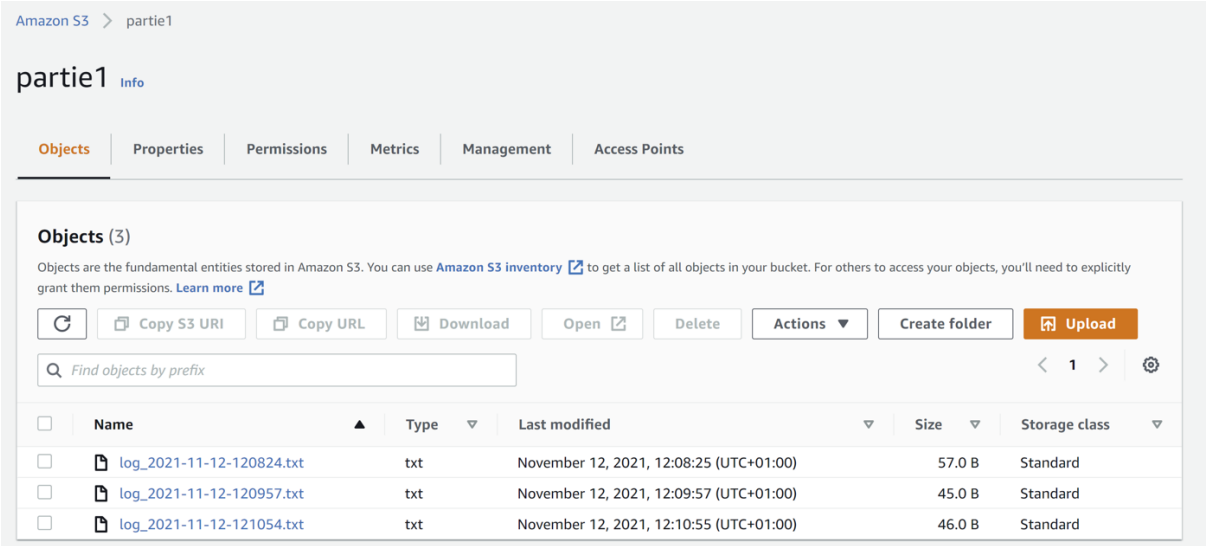
Finally, the client reads the SQS queue response and prints the results.

Different scenarios :

- ☐ All positive integers, len(list) <= 10 : returns expected results : min, max avg and median.
- ☐ One or more negative numbers : an error message stating that all numbers should be positive.
- ☐ len(list) > 10 : an error message stating that the list should contain no more than 10 numbers.



Respective log files :



Logs' contents :

You can find a video demo explaining further the application's process in this link :
https://www.youtube.com/watch?v=jh1teJYidpQ&ab_channel=MohammedBenmoussa

The full code for the application is in the .zip enclosed with this paper.

Github link :
https://github.com/MohammedBenmoussa/projet-cloud/tree/main/partie_1

Go Further: the hot-dog application :

We're working with the following scenario : we work for a very promising company whose goal is to develop an app capable of recognizing hot-dogs on images. To proceed, the company needs labeled data, meaning data with a label 1 when it represents a hot-dog ; 0 when it represents something else. To help collect and annotate data, we will create a new AWS application that shows a sample of images to a client (each image successively), let the client label each of the images, and record/store that label in a table.

• The set of images available for labeling is stored in a S3 bucket

• The file storing the label information for each image is also on a S3 bucket

• Each client is shown 4 random images out of the total number of images that are in the bucket

• If multiple client are shown the same image, you should record each answer in the same text file

• A client can contribute and upload more hot-dog images to the bucket

• We upload around 20 images, half of these images represent hot-dogs. As for the other half, they’re random pictures.

Our solution :

•We chose python as our programming language.

We have a total of 20 pictures stored locally. Half of these pictures are hot-dogs...

... the other half are not.

Our application is divided into two parts, the admin part...

(command : **python.exe .\admin.py**)

... and the user one.

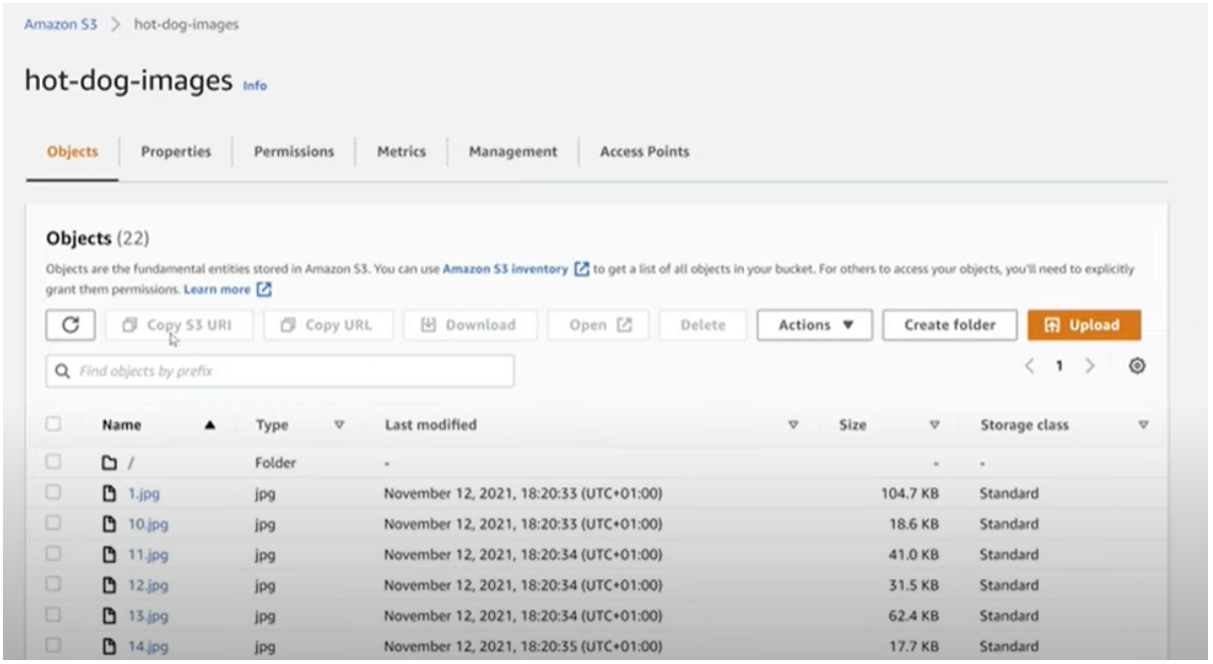
(command : **python.exe .\client.py**).

Each part is responsible for executing certain tasks.

1 - Uploading the images to the S3 bucket :

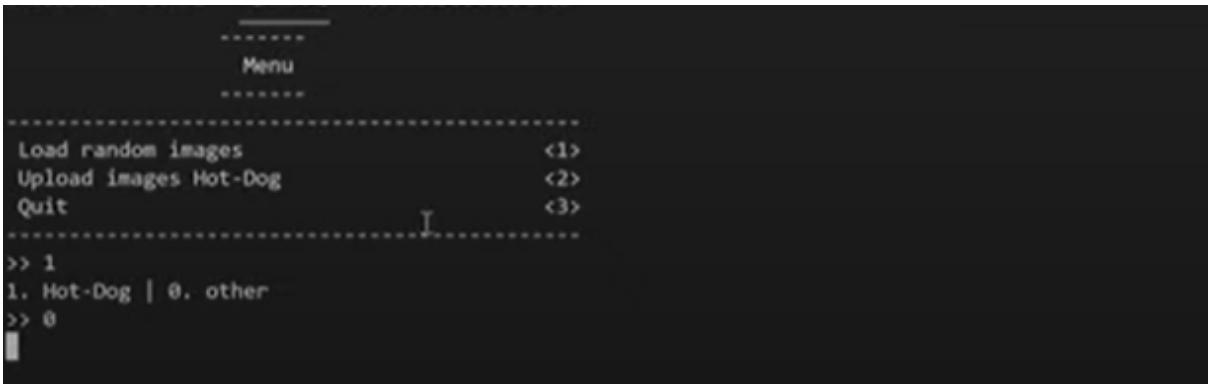
The admin / server is responsible for this task.

The 20 images are initially stored locally. We choose the option “Upload images” to upload them in the S3 bucket.



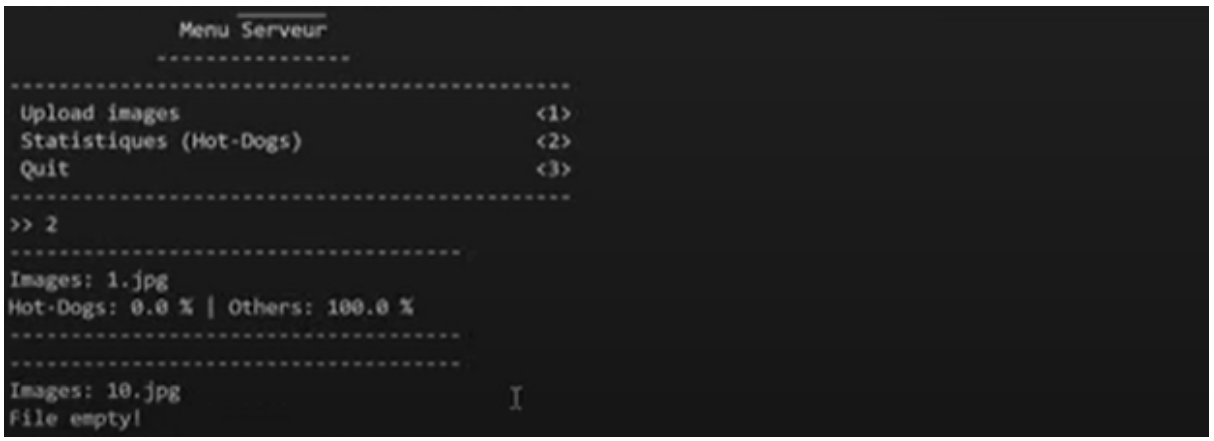
2 - Loading random images :

The user / client is then presented with 4 random images. After being shown an image, he must write whether it is a hot dog (1) or not (0).



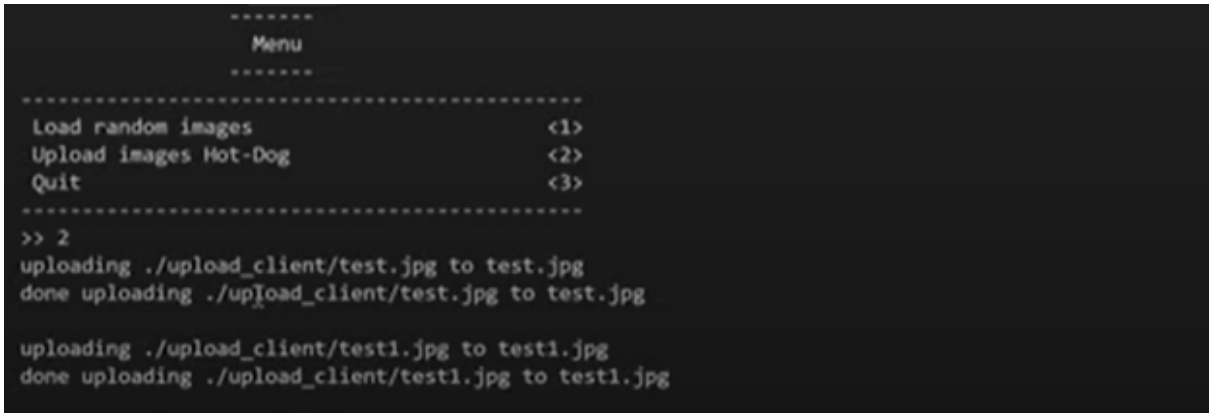
3 - Statistics :

The answer to each image is stored in a file that is rendered by the admin for statistics’ purposes :



4 - User uploading images :

The user can also upload his own images into the S3 bucket :



You can find a video demo explaining further the application's process in this link :
https://www.youtube.com/watch?v=HAULDuyIkdM&ab_channel=MohammedBenmoussa

The full code for the application is in the .zip enclosed with this paper.

Github link :
https://github.com/MohammedBenmoussa/projet-cloud/tree/main/partie_2