# Lab Work 2 - Cloud Computing

Charlotte Laclau
prenom.nom@univ-st-etienne.fr

October 8, 2021

## Instructions

- This lab is your second evaluation for this course
- You can form groups of **maximum 4 students**
- Each group should share a link to a github repository.
- The repository will include all the files to run your code, A README, a small report describing your project and a 3 minutes maximum demo video of your code.
- **Deadline : October 31th 2021**

## Goals

- Create an AWS application in which a client submits a request to a worker on EC2. The request is composed of a list of integers.
- The EC2 worker is a Python application that receives the list (the request) and calculates the min, max, mean, median and send them back to the client.
- After sending the response, the EC2 worker must write a log file describing this transaction as txt file on S3 Amazon Storage Cloud.
- Additional details
  - the client is limited to 10 numbers. If more numbers are entered, a message error should be sent
  - numbers must be positive - again a message error is sent to the client if one of the number is not positive

## Required Tools

To proceed with the lab, you will need the following tools. Make sure that you have Python installed and the associated libraries. Other services are included in AWS.

- Python 3 with the library **boto** or Eclipse with AWS plugin installed if you want to do the project in Java..
- Amazon EC2
- Amazon S3
- Amazon SQS (Simple Queue Service)

What is Amazon SQS ? What type of queues can you create ? State their differences.

## Introduction : Sample Tutorial on boto3

In order to get familiar with the use of the boto library, follow the **Sample Tutorial**.

Before starting make sure that you configured properly boto with AWS (See Quickstart → Configuration).

What is the difference between a resource and a client ?

## Project Architecture

The project has a classic Web-Queue-Worker architecture. The core component 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**.

In which situations is a Web-Queue-Worker architecture relevant ?

More precisely -

1. **Client** : a java/python AWS project (with one class that includes main method). It is hosted on your machine (not on the cloud). This client does the following steps:
    - Creates an SQS Queue for requests (called the requestQueue) and use it to submit a request containing a list of maximum 10 integers (separated by comments).
    - Receives the response from the EC2 worker.
2. **EC2 Worker**: this is a java/python application that runs on an EC2 instance. It does the following tasks:
    - Reads the messages from the requestQueue
    - Calculates the results
    - Creates a response queue (called responseQueue) and submits the result on it.
    - Create a new log file (a text file) and store it on Amazon S3.

## Go Further: the hot-dog application

You work for a very promising company which goal is to develop an app capable to recognize 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, you 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 image, 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
- You are free to choose whatever images you want to build the set of images (you need around 20 images). Half of this images must represent hot-dogs. As for the other half, pick whatever you want (let's all stay correct).
- Choose images of small size

Good luck !