Télécom Saint-Étienne
Cloud Computing
Working With Amazon AWS

Enseignant: Amro Najjar

Remarks

- You need to form groups of two students (binôme).
- A binôme should write a report containing the answers of questions and exercises listed in this document and send them to amro.najjar@telecom-st-etienne.fr.
- **You can write the report in French**.

1 Objectives

- 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 Java 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.

2 Tools Needed

- Eclipse with AWS plugin installed.
- Amazon EC2.
- Amazon S3.
- Amazon SQS (Simple Queue Service)

3 Project Architecture

1. **Client**: a java 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:
   (a) Creates an SQS Queue for requests (called the requestQueue) and use it to submit a request containing a list of 20 integers (separated by comments).
   (b) Receives the response from the EC2 worker.
2. **EC2 Worker:** this is a java application that runs on an EC2 instance. It does the following tasks:
   (a) Reads the messages from the requestQueue
   (b) Calculates the results
   (c) Creates a response queue (called responseQueue) and submits the result on it.
   (d) Create a new log file (a text file) and store it on Amazon S3.

Figure 1 shows on overview of the architecture.

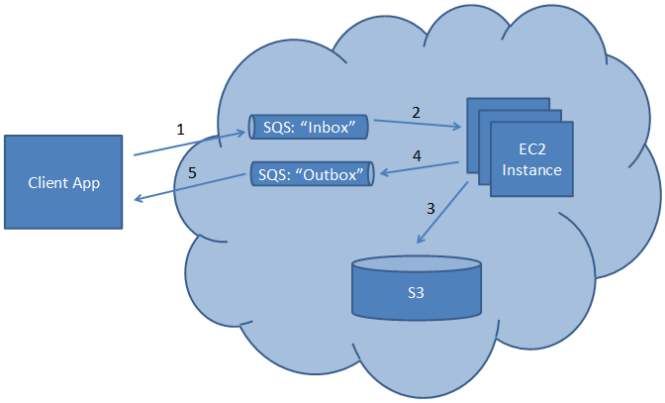

FIG. 1 – *Overview of the architecture. Source: a project prepared by Avener May in the University of Columbia, NY, USA.*

4 Amazon SQS

Amazon SQS is a queueing service provided by Amazon. It make it possible for different AWS components (e.g. EC2 and S3 instances) to connect and exchange messages.

```
Question 1:
-Explain what is SQS and why it is used
-What is Amazon SES?
```

5 The Client Project

This section will guide you to create the client project which is hosted locally on your own machine.

- Open Eclipse EE (the one where you installed AWS plugin) ►`New` ►`Other` ►`AWS` ►`AWS Java Project` ►`Next`.
- Un-check all the check boxes and name your project `TD2` ►`Finish`.
- This project should contain one class with a main method.
- This class should do the following steps:
  1. Create a String composed of a list of 20 integers separated by commas.
  2. Create a SQS queque called *requestQueue* in which the client submits its requests.
  3. To create a SQS queue, you can follow the instruction found in the sample project called `Amazon Simple Queue Service Sample`. To get the code of this sample, create a new AWS java project and select the `Amazon Simple Queue Service Sample`.
  4. This project contains only one Java class named `SimpleQueueServiceSample` that provides an example of creation and utilization of a SQS queue. For this reason, this file is an important source of ideas for today's TD. Read the comments carefully.
  5. This class should also receive the result from the `ResponseQueue` created by the EC2 worker (See Section 6).
  6. **Important Remarks**:
     (a) Since you want to minimize the costs of your resources, you should be sure that you create only one RequestQueue and only one ResponseQueue. Creating more queues may lead to unnecessarily costs paid to Amazon.
     (b) Sometimes, when an methods wants to read the content of queue, but the queue maybe still empty. For this reason, it is better to make the method wait a bit before raising an exception about the queue being empty. To do so, you should

change the `ReceiveMessageWaitTimeSeconds` to 10 seconds (its default value is 0). Figure out how to that in your code.

6 The EC2 Worker Project

- This project is an `AWS Java Project`. It contains a server that will be later migrated to an EC2 instance.
- Create a class that contains a main method. This class should perform the following tasks:
  1. Create the `ResponseQueue`.
  2. Get the client requests from the requestQueue, calculate the result and post it in the response queue.
  3. Create a file on Amazon S3 where you store a log for this transaction (request message, response message, Time-stamp, IP of the client, etc).
  4. Create a new EC2 instance and copy your EC2 worker project on the new instance and run it (as we did in TD1).
- **Important Remarks**
  - Since you want to minimize the costs of your resources, you should be sure that you create only one RequestQueue and only one ResponseQueue. Creating more queues may lead to unnecessarily costs paid to Amazon.
  - In order to understand how S3 works, your check out the Sample S3 tutorial project. To do so, go to `Eclipse ►New ►Other ►AWS Java Project` and select `Amazon S3 Sample`. It might also be useful to see `Amazon S3 Transfer Sample.`.
  - Remember you only need to create one bucket. Inside it, you can put several txt files.

7 Go Further: Image Processing Application

In this part, we will create a new AWS application that re-sizes (or in some other way processes) an image which a user uploads to the cloud. You should make the following changes:

- Now the client uploads images to bucket in S3, places the "key" to this uploaded image in the "inbox" queue, and waits for this message to be processed. Once this message is processed, a message will be placed in the "outbox" queue with the "key" to the resized image. Once this message arrives, the Client App should download the resized image from S3. (make sure to correlate between requests and replies, as in Part 1).
- SQS request/response queues: same as we did in the first part.
- 2+ EC2 worker instances: These instances wait for messages to appear in the "inbox" queue, and when they do, they retrieve the referenced image file from S3, perform some image- processing on the image, place the resulting image in S3, place the "key" to the "result" image file in the "outbox" queue, and delete the message from the "inbox" queue.
- S3 bucket: This will contain the original and processed images.
- **Important Remark**:
  1. Choose a small size image (e.g. 50 KBytes).
  2. Your instance is running on Amazon Linux OS. Under this OS, you can make simple image processing operations using the ImageMagick library. To install it, you only have to execute the following command `sudo yum install ImageMagick`. When ImageMagick is installed, you can use the convert command. `convert` can do many image manipulation operation. You can see its documentation on this link. Choose the convert function that you like. If you do not want to choose, you ask the client to choose the convert operation he wishes in his initial request.

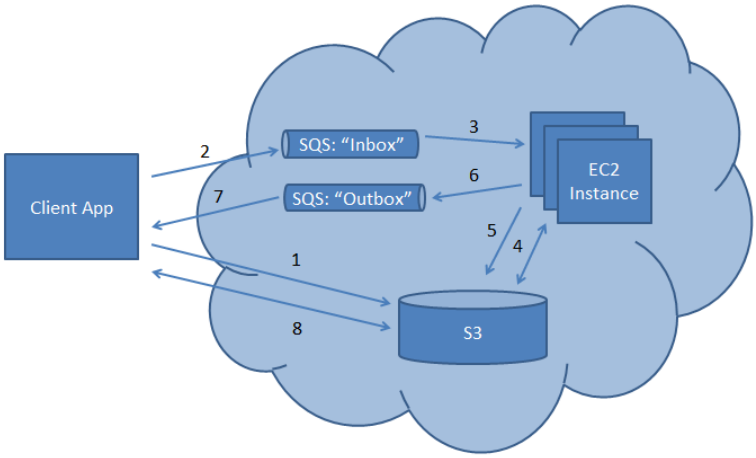

FIG. 2 – *Overview of the new architecture. Source: a project prepared by Avener May in the University of Columbia, NY, USA.*

3. To automate the conversion process, you should put the ImageMagick commands in java code.

Figure 2 illustrates the new architecture.