Distributed systems assignment 1 README

Client: The following is the usage print when running the client jar with '-h'.

It explains everything that you need to know to run the client. We recommend you run the jar once with '-h' for the first time to generate the input/output folders

```
C:\Users\Yuval\Desktop>java -jar client.jar -h
Usage: java -jar clientProgram.jar [optional quick start args] [-h | -help]
                                   [-d] [optional debug flags]

-h | -help :- Print this message and exit.

-d | -debug :- Run in debug mode, logging all operations to standard output.

optional quick start args:

    inFileName1... inFileNameN outFileName1... outFileNameN n [terminate]

    n :- Reviews per worker.

    terminate :- send terminate signal.

optional debug flags:

    -ul | -uploadLog :- Ec2 instances will upload their logs to the S3 bucket.
                Must be used with -debug.

    -ui | -uploadInterval <interval in seconds> :- When combined with -uploadLog, specifies the interval in seconds
                between log uploads to the S3 bucket.
                Must be a positive integer, must be used with -uploadLog.
                If this argument is not specified, defaults to 60 seconds.
                Minimum interval is 10 seconds.

    -noEc2 :- Run without creating worker instances. Useful for debugging locally.

    -noManager :- Run without creating manager instance. Useful for debugging locally.
                All other debug flags are ignored when this flag is used.

credentials for aws:

    The credentials file must be in the same directory as the jar file,
    should be named 'credentials.txt' and contain the following:
            aws_access_key_id = <your access key>
            aws_secret_access_key = <your secret key>
            aws_session_token = <your session token>

input / output files:

    input files should be placed in the 'input_files' directory,
    output files will be placed in the 'output_files' directory.
    both directories will be automatically created in the same directory as the jar file
    upon running the program for the first time (or when running with -h | -help)
```

You can run the client with the arguments as required by the assignment instructions or you can run the client without any arguments and head directly to the main menu. We call the first option "quick start"

```
C:\Users\yuval\Desktop>java -jar clientProgram.jar
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.

Choose an option:
1. Send new request
2. Show requests
3. Open finished request
4. Exit
>>
```

Manager: The manager does not require any special arguments outside of debug args if you wish to use them.

```
C:\Users\Yuval\Desktop>java -jar manager.jar -h
Usage: java -jar managerProgram.jar [-h | -help] [optional debug flags]

-h | -help :- Print this message and exit.

optional debug flags:

    -d | -debug :- Run in debug mode, logging all operations to standard output

    -ul | -uploadLog <file name> :- logs will be uploaded to <file name> in the S3 bucket.
                Must be used with -debug.

    -ui | -uploadInterval <interval in seconds> :- When combined with -uploadLog, specifies the interval in seconds
                between log uploads to the S3 bucket.
                Must be a positive integer, must be used with -uploadLog.
                If this argument is not specified, defaults to 60 seconds.

    -noEc2 :- Run without creating EC2 instances. Useful for debugging locally.

credentials for aws:
    The program will use the default aws credentials provider chain to get credentials.
    We recommend using environment variables to set credentials.
```

Worker: The worker has a lot of mandatory command line arguments. This allows the workers to be configurated at runtime and makes them flexible and scalable.

```
C:\Users\Yuval\Desktop>java -jar worker.jar -h
Usage: java -jar managerProgram.jar -workerId <id> -inQueueUrl <url> -outQueueUrl <url>
        -managerQueueUrl <url> -S3BucketName <name> -timeout <visibility timeout in seconds>
         [-h | -help] [optional debug flags]

-h | -help :- Print this message and exit.

optional debug flags:

    -d | -debug :- Run in debug mode, logging all operations to standard output

    -ul | -uploadLog <file name> :- logs will be uploaded to <file name> in the S3 bucket.
                Must be used with -debug.

    -ui | -uploadInterval <interval in seconds> :- When combined with -uploadLog, specifies the interval in seconds
                between log uploads to the S3 bucket.
                Must be a positive integer, must be used with -uploadLog.
                If this argument is not specified, defaults to 60 seconds.

credentials for aws:
    The program will use the default aws credentials provider chain to get credentials.
    We recommend using environment variables to set credentials.
```

Instance types used:

Worker – t2.large

Manager – t3.micro

How long it took to run: ~11 minutes

n used : 5

NOTE FOR RUNNING WORKER & MANAGER:

As mentioned in the usage, we recommend setting the credentials in the environment variables. As such, we bundled a small tool that helps with that. In order to use it, you put the credentials in the 'credentials.txt' file that is located in the same folder as the jar and it reads the credentials from there and puts them into the environment variables.