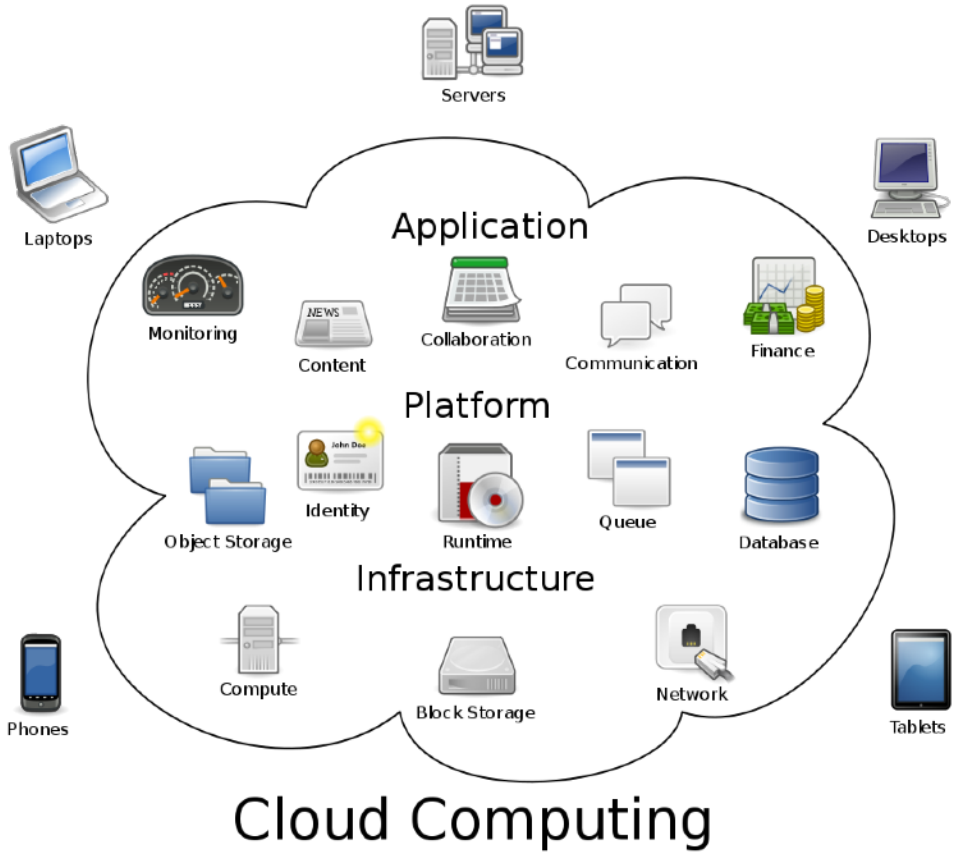
Assignment 4
AWS SQS, EC2, and S3

COMP3358 Distributed and Parallel Computing

A basic application with SQS, EC2, and S3

- For this part of the assignment, you will create an AWS application which resizes an image which a user uploads to the cloud



AWS SQS

- Amazon Simple Queue Service (SQS) is a fully managed message queuing. SQS offers a reliable, highly scalable, and cost-effective way to send, store, and receive messages between software components.

- A bit more details about SQS: https://aws.amazon.com/sqs/
- https://www.youtube.com/watch?v=CyYZ3adwboc

Client app

- Client app 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. (Note: it is important that the client only pay attention to the message which is the response to the request it sent.)

SQS queue

- SQS request queue (“inbox”): This is where requests are stored until they are processed
- SQS response queue (“outbox”): This is where responses are stored until the user sees the response to his message

EC2 worker instances

- EC2 worker 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
  - With regard to how to perform image processing: See Linux “convert” command, which you can install via the command “sudo yum install ImageMagick” on your EC2 machine. Pick some function of “convert” which you like, and have your EC2 machines perform this function on the uploaded images. Suggestions include: resize, paint, etc

```
sudo apt update
sudo apt install --no-install-recommends imagemagick
```

S3 bucket

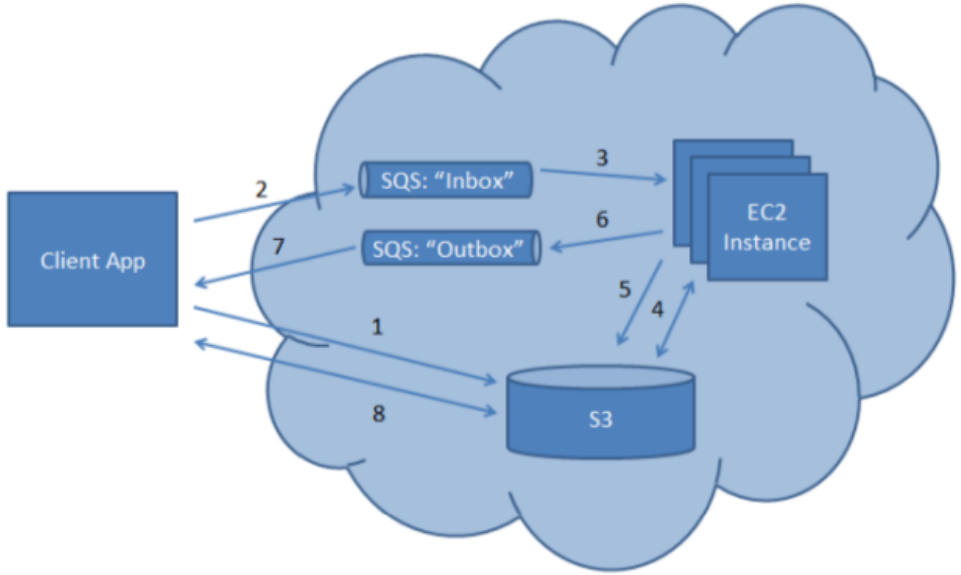
- S3 bucket will contain the original and processed images

AWS SDK for Java

- Code Examples (only the code needed to demonstrate each technique, like creating, listing, and deleting Amazon S3 Buckets):
  - Amazon S3 Examples: https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/examples-s3.html
  - Amazon SQS Examples: https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/examples-sqs.html
  - Amazon EC2 Examples: https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/prog-services-ec2.html

Submission requirements

- The architectural diagram of the application is shown below



- Specify the each steps in the architecture above (with used AWS APIs), 30%
- Submit all of your source code to Moodle, with an experiment report. The report should include the code structure of your project, commands to run your code, and screen capture after you running each steps according to the diagram.