

Amazon EC2

Name: Shuja Khalid

Email: shuja.khalid@mail.utoronto.ca

Date: 19 March 2017

1. Purpose:

The purpose of the project is to create a scalable cloud-based web application that can automatically scale itself to adjust to fluctuating user demand. Although Amazon already provides such services, this application provides insight about the level of detail that is required to construct and maintain truly scalable applications. The application should also have the ability to apply transformations to user defined images using the Python based tool ImageMagick.

2. Methodology:

The project included a set of strict guidelines but also provided some flexibility in the approach to create the application. The flowchart depicted in Fig. 1 shows illustrates the process that was chosen.

Key Features:

- I. The MySQL database was hosted on a separate instance and consisted of a user credentials table and an image information table
- II. The user credentials table consisted of the user's login and password whereas the image information table consisted of the name of an uploaded image along with keys for each of the image transformations (90 deg. Rotations)
- III. A background task was run using an existing python library (APScheduler) to monitor the average CPU Utilization of the available instances and to increase or decrease the worker load accordingly (depicted in red in Fig. 1)
- IV. AP Scheduler runs in standalone mode to ensure that multiple instances aren't running on multiple threads at once. To stop the scheduler, type "Isot -i" and kill the process associated with port 5000 using "kill PID"
- V. The application accepts user defined images and transforms them by rotating the original image in increments of 90 deg.
- VI. The application stores the images uploaded by the user and the relevant transformations in Amazon S3 storage
- VII. The administrator interface allows complete control over the application (Auto-scale setup, Manual Instance Launch, Memory Purge, Instance Monitoring)
- VIII. Checks and balances have been included to ensure that input values provided by the user are acceptable (i.e. password and username are at least 4 characters, passwords match during login)

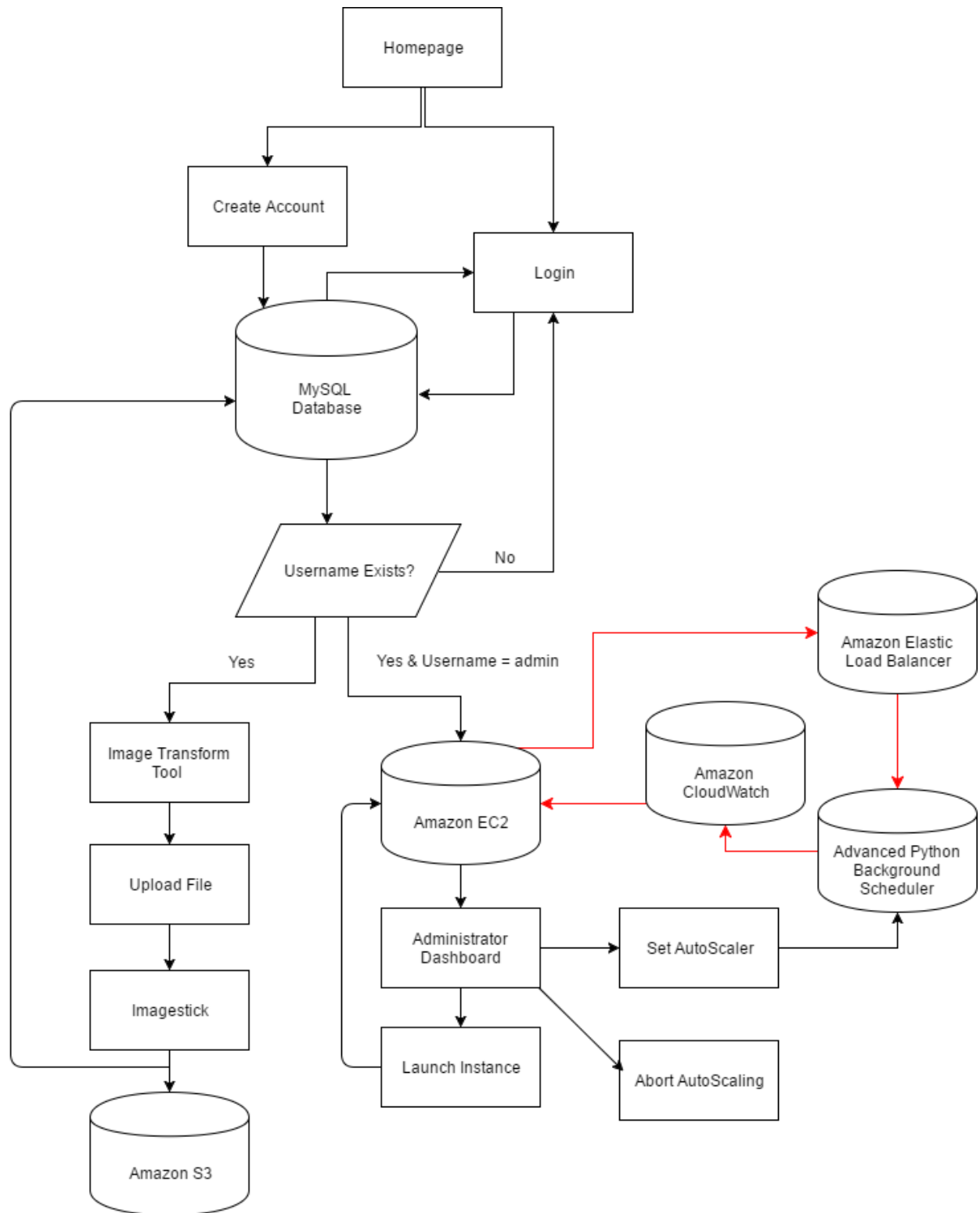


Figure 1: Application Process Flowchart

4. Code Structure and required files:

The code consists of two AMIs:

The code associated with the browser based interface (manager and user) and the code associated with the workers that are used in case the CPU Utilization values exceed the provided threshold values is present in the following AMI:

5. Launching the Manager Instance using an AMI:

- i.) Launch an instance
- ii.) Select t2.small general purpose instances

<input type="checkbox"/>	General purpose	t2.micro Free tier eligible
<input checked="" type="checkbox"/>	General purpose	t2.small
<input type="checkbox"/>	General purpose	t2.medium

Next: Configure Instance Details

iii.) Choose

iv.) Select the following checkboxes:

Enable termination protection ⓘ

☒ Protect against accidental termination

Monitoring ⓘ

☒ Enable CloudWatch detailed monitoring

Next: Add Storage

v.) Choose:

Next: Add Tags

vi.) Choose:

Next: Configure Security Group

vii.) Choose:

viii.) Choose an existing security group:

Assign a security group: ☐ Create a **new** security group

☒ Select an **existing** security group

ix.) Select a pre-existing security group:

☒ sg-f93ed886 launch-wizard-5

The group should have the following rules:

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ
All traffic	All	All	0.0.0.0/0
All traffic	All	All	::/0
SSH	TCP	22	0.0.0.0/0
Custom TCP Rule	TCP	5900 - 5902	0.0.0.0/0
MYSQL/Aurora	TCP	3306	0.0.0.0/0
MYSQL/Aurora	TCP	3306	::/0

Review and Launch

- x.) Choose:
- xi.) When prompted for a new key pair, use the provided pem file (ece1779_a1.pem)
- xii.) Click “Launch Instances”

6. Running the User/Manager Instance:

- xiii.) Once the manager instance is created, use ece1779_a1.pem to ssh into the instance using the following command:

```
$ ssh -i .ssh/ece1779_a1.pem ubuntu@{Insert Public IP Address here} -L 5901:localhost:5901
```

- xiv.) Optional: use VNC Viewer to visualize the instance
- xv.) Open a terminal and run the manager start up script using “launch_admin”. This script runs the application on port 5000
- xvi.) Open firefox inside the instance if using VNC viewer or open up a browser on your computer and in the address bar type the following address and hit run:

<http://{Insert Public IP Address here}:5000>

7. Proceeding as a user:

- xvii.) Use the following credentials to proceed as a user:
Username: Superman
Password: Batman

An account can be also created on the web application to continue as a user and proceed to use the image transformation tool

8. Proceeding as the Manager:

- xviii.) Use the following credentials to proceed as the Manager:
Username: admin
Password: yoyo

This view can be used to launch instances, set up auto-scaling, view CPU utilization of existing instances, delete instances and purge all user data (images table from database + content in S3 bucket).

9. Testing using the Load Generator:

- xix.) Run “launch_admin” after ssh’ing into the instance. This terminal will contain a progress report of the Auto-scaler after the auto-scaler is run in the background (Figure 2)
- xx.) Run “launch_user” in another terminal. This terminal will show file upload progress once the load generator is run (Figure 2)
- xxi.) From the terminal, run the following commands to set up the load generator:

```
cd /home/ubuntu/Desktop/ece1779/loadgenerator
java -cp . ece1779.loadgenerator.LoadGenerator {Insert_Public_IP_Address_here} 5001
superman batman
```

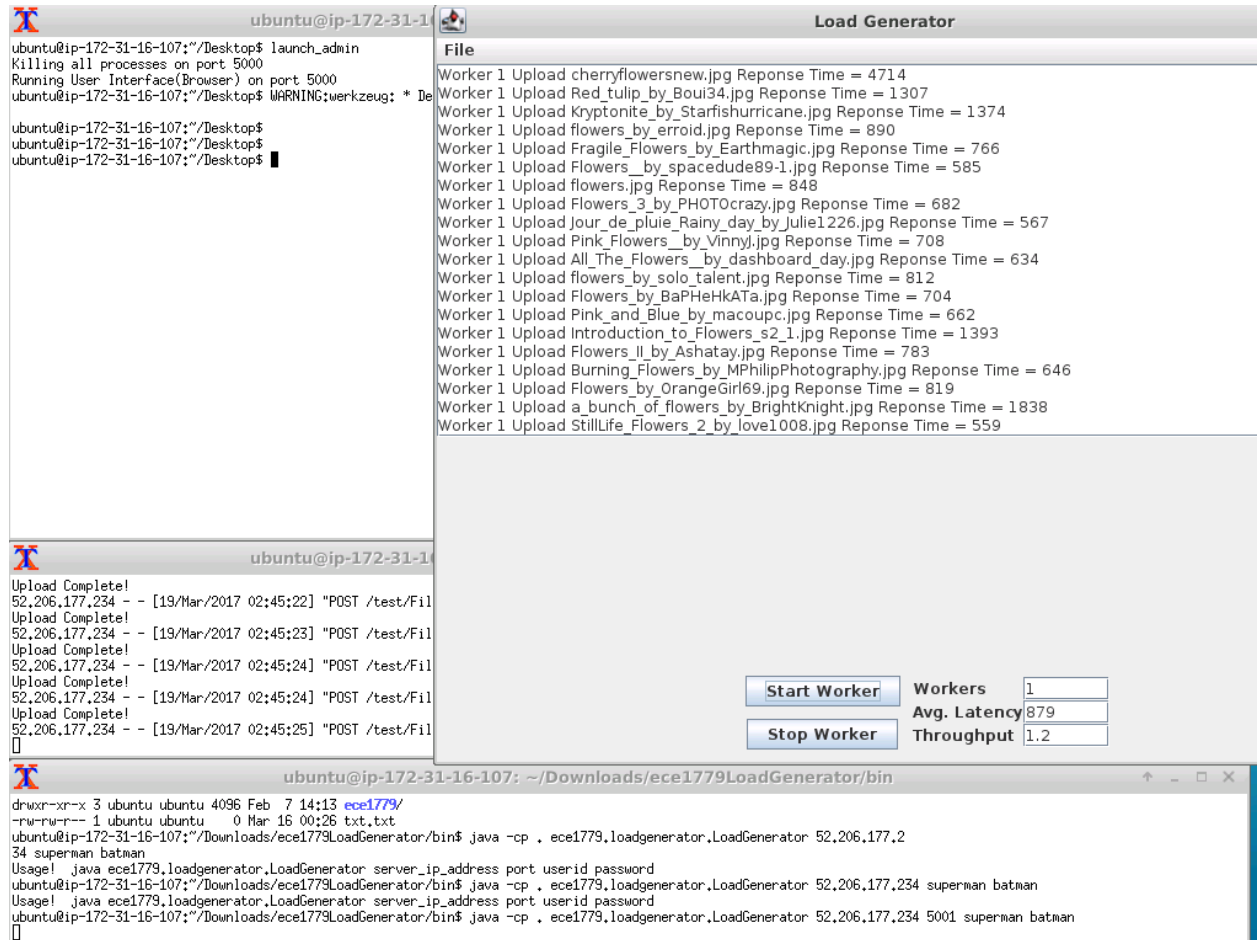


Figure 2: Test Setup to run the load generator using VNC Viewer

Note: Testing has been done using at max 4 workers that generate a throughput of ~4.5. Higher values resulted in Amazon S3 errors as only 4 images can be uploaded at a time to the S3 bucket. This error was surprising and a decision was made to remain within the limits to err on the side of caution.

10. CloudWatch Monitoring and Auto-Scaling:

The Auto-scale Configuration option from the Administrator Dashboard can be used to set threshold values and multipliers for the auto-scaling process. A Background Scheduler is launched which monitors the CPU utilization every 5 seconds. This Displays

Database Attributes:

A separate instance was used to host the MySQL database. The following tables were created in the database as per the requirements:

users		
Name	Type	Description
id	int	User ID (autoincrement)
login	varchar(16)	User login
password	varchar(16)	User password

Table 1: User Attributes

images		
Name	Type	Description
id	int	Image ID (autoincrement)
userId	int	User ID
key1	varchar(255)	S3 key of original image
key2	varchar(255)	S3 key of 1st transformation
key3	varchar(255)	S3 key of 2nd transformation

Table 2: Image Attributes

11. Web Application layout:

The web application was designed to keep the user interface simple and interactive. jQuery libraries such as alertify, fancyInput and tablesorter were used to improve the quality of the interface. Alertify was used to make custom notifications, fancyInput created interactive input fields and table sorter was used to sort table columns in the administrator dashboard.