# Capstone Project

## 1.Notes:

### 1.1 Subdomains

https://honglibu.ucmpcs.org

### 1.2 Check file size in Ex.2

I implemented file size check using **javascript**. Firstly, I add a parameter called **'size'** when return render_template in function annotate() in views.py. Then, I add javascript in annotate.html. In the script, the file size will be checked immediately after user clicking 'Annotate'. If free user are uploading a file with size larger than **150KB**, I will redirect user to subscribe page, otherwise, the file will be uploaded successfully.

### 1.3 Archive free user date to Glacier in Ex.7

I accomplished this task using **SQS and SNS**. The job_archive queue listens to the job_result topic. The notification message will be sent to this queue every time  a job is completed. The **results_archive.py** is runs on util instance. The script keeps polling the queue. When it receives a message, role in the queue message will compare with **session['role']:**

1) If the user is free user, it will calculate the time difference between completed time of the job and time().time. If the job is completed within 30 minutes, do nothing. Otherwise, if the job has been completed for more than 30 minutes, the result file will be archived to glacier, and the archive ID is record in DynamoDB.
2) If the user is a premium user, it will delete the message.

I chose SQS method because it is easy to implement, and I can dump all useful information in the queue, so that it won't waste resources to populate scan DynamoDB.

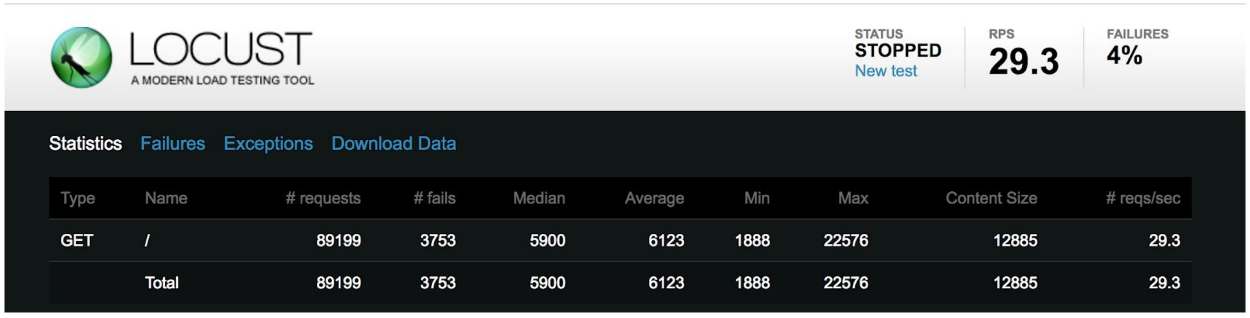
### 1.4 Restore data for premium user in Ex.9:

Firstly, I initiate Glacier archive job when use successfully upgrade. Then I send a notification to a **SQS** called **job_restore** through corresponding **SNS topic** when a job is successfully initiated. The python script archive_restore is running on util instance. It keeps polling the queue. When it receives a message, it will restore the data to S3, and delete the message.

### 1.5 Load test on web servers in Ex.13:

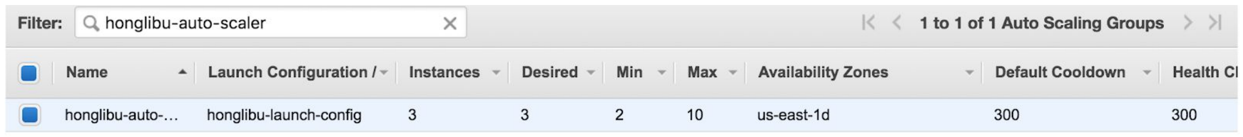
The instances incremented by 1 every 5 minutes.  It's because the load balancer kept receiving high-load requests per minute. So the state of the alarm

“honglibu-web-scale-out” kept being “ALARM”. And,I had set scale-out policy as “300 seconds to warm up”. So the auto scaler would wait for 5 minutes before adding a new instance. When the instance number reached the maximum number of instances we set, the increment stopped.
Here is my locust screenshot when instance number reached the maximum number:

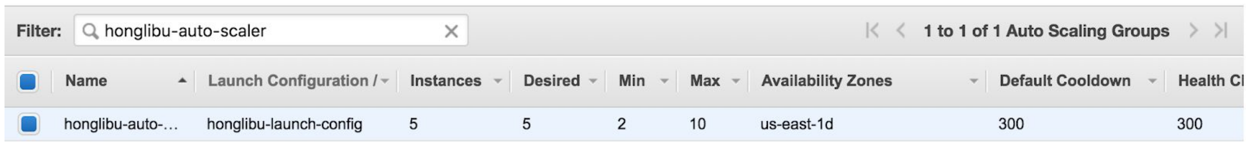


| Type | Name | # requests | # fails | Median | Average | Min | Max | Content Size | # reqs/sec |
|---|---|---|---|---|---|---|---|---|---|
| GET | / | 89199 | 3753 | 5900 | 6123 | 1888 | 22576 | 12885 | 29.3 |
| | Total | 89199 | 3753 | 5900 | 6123 | 1888 | 22576 | 12885 | 29.3 |

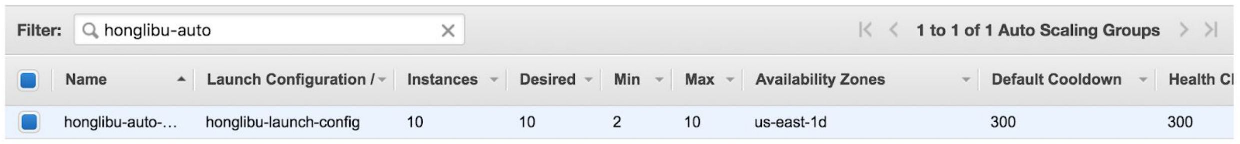
Below several screenshots show the stage when scaling out. At the beginning, two instances is running and ultimately, the instances reach 10.



| Name | Launch Configuration / | Instances | Desired | Min | Max | Availability Zones | Default Cooldown | Health Cl |
|---|---|---|---|---|---|---|---|---|
| honglibu-auto-… | honglibu-launch-config | 3 | 3 | 2 | 10 | us-east-1d | 300 | 300 |



| Name | Launch Configuration / | Instances | Desired | Min | Max | Availability Zones | Default Cooldown | Health Cl |
|---|---|---|---|---|---|---|---|---|
| honglibu-auto-… | honglibu-launch-config | 5 | 5 | 2 | 10 | us-east-1d | 300 | 300 |

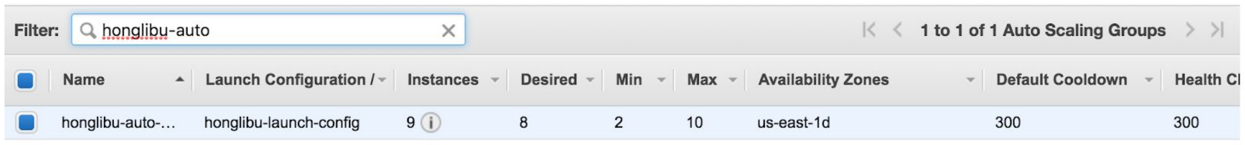


| Name | Launch Configuration / | Instances | Desired | Min | Max | Availability Zones | Default Cooldown | Health Cl |
|---|---|---|---|---|---|---|---|---|
| honglibu-auto-… | honglibu-launch-config | 10 | 10 | 2 | 10 | us-east-1d | 300 | 300 |

After I stop Locust test, web farms start to scale in according to my ‘honglibu-web-scale-in’ policy. In this policy i defined the threshold as ‘TargetResponseTime < 0.05  for 1 datapoint within 1 minute’. So if I stop the load test or reset the load test to be in low request workload(e.g. 1 user, at 1/sec), web response time will take less than 50ms, then the ELB will coordinate Auto Scaling group to terminated  instances by 1 every 5 minutes. Of course, if consider server latency, the scale in alarm might be bounce between ‘ok’ and ‘alarm’.

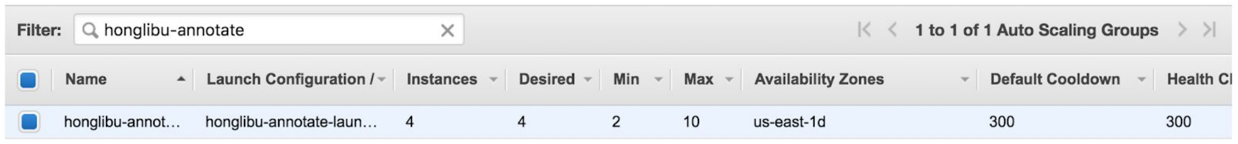
### 1.6 Load test on anntools servers in Ex.14:

Here is my screenshots of anntools auto scaling group when i was running autotest script in my local machine. As the anntools alarm, when i continuously send job requests, the sum  of job in 15 minutes will reach 30(Threshold). Consequently, the anntools farm will scale out by increasing 1 instance per 5 minutes. When I stopped, the message received will be decreasing, and then the anntools farm will scale in. Finally, there will still have two instances running.



Filter: honglibu-auto — 1 to 1 of 1 Auto Scaling Groups

| Name | Launch Configuration / | Instances | Desired | Min | Max | Availability Zones | Default Cooldown | Health Cl |
|---|---|---|---|---|---|---|---|---|
| honglibu-auto-… | honglibu-launch-config | 9 | 8 | 2 | 10 | us-east-1d | 300 | 300 |



Filter: honglibu-annotate — 1 to 1 of 1 Auto Scaling Groups

| Name | Launch Configuration / | Instances | Desired | Min | Max | Availability Zones | Default Cooldown | Health Cl |
|---|---|---|---|---|---|---|---|---|
| honglibu-annot… | honglibu-annotate-laun… | 4 | 4 | 2 | 10 | us-east-1d | 300 | 300 |

## 2. Instructions:

2.1 After uploading files , it might take minutes to run annotator, so be patient please.
2.2 when you need to logout and login, please logout 'UChicago' account if possible.
2.3 When you restore a job result, you need several minutes to download files.

## 3.Reference:

- 1. Java script to check file size: https://stackoverflow.com/questions/3717793/javascript-file-upload-size-validation
- 2.DynamoDB query: https://stackoverflow.com/questions/34171563/how-do-i-query-aws-dynamodb-in-python http://boto3.readthedocs.io/en/latest/reference/services/dynamodb.html#table
- 3. Python change time format: https://stackoverflow.com/questions/19825330/python-how-to-convert-ctime-to-m-d-y-hms
- 4. Generate presigned url: http://boto3.readthedocs.io/en/latest/reference/services/s3.html#S3.Client.generate_presigned_url
- 5.s3 object read: https://stackoverflow.com/questions/31976273/open-s3-object-as-a-string-with-boto3/35376156
- 6. Stripe create a cumster: https://stripe.com/docs/api#customer_object

- 7. Glacier initiate job: http://boto3.readthedocs.io/en/latest/reference/services/glacier.html#Glacier.Client.initiate_job
- 8. Python configparser: https://docs.python.org/3/library/configparser.html
- 9. Ses send email: https://boto3.readthedocs.io/en/latest/reference/services/ses.html#SES.Client.send_email
- 10. Glacier upload archive: http://boto3.readthedocs.io/en/latest/reference/services/glacier.html#Glacier.Client.upload_archive
- 11. Glacier get job output: http://boto3.readthedocs.io/en/latest/reference/services/glacier.html#Glacier.Client.get_job_output
- 12 stripe test cards: https://stripe.com/docs/testing
- 13 install simplejson: https://stackoverflow.com/questions/718040/how-to-install-simplejson-package-for-python