

Name: Martin Wong

Student ID: s3833684

COSC2737 Assignment 2 IT Infrastructure in the Cloud

1 Rationale:

Due to current pandemic situation, gathering is banned for any circumstances. University has changed the teaching mode to remote teaching. Although it is inconvenient that face-to-face teaching is not available, the remote teaching mode still provide some flexibility to student to learn. Students can choose when they watch the lecture video but not attend the lecture at a fixed place at a fixed time. Also, the technology can help the learning process to some extent. This project aims to use the technology provided by AWS to help university students for the current social situation.

The project is building a video player on a web server, that can have faster playback speed like 1.25x or 1.5x etc. Students can watch the video at a faster speed and consume less time. However, faster audio speed may cause difficulty for students to hear and understand the content. Therefore, subtitles are required. Finally, the generated subtitles can be downloaded by the students as a study material. Though the lecture notes are supposed to contain all the content of the lecture, it may not be as detail as what the lecturer said. The transcript will record every single word of lecture into a text file and students can easily find specific content by using search function which is much more time efficient than searching specific content in a video.

The project applies three AWS tools: EC2, S3 and transcribe. Firstly, EC2 hold a http web server for users to access the content of the project. Apache2 is applied for the web server. Flask framework, which was written in python, is adopted for the web. There are three reasons I chose flask. The first one is I learn python before and I found it easy to learn and consume less time to debug and set up. The second reason is that AWS supports boto3 library in python to control all the services. The third reason is there are lot of online tutorial and learning material about flask and boto3. Wsgi is applied to connect the flask app and apache2 server. The second AWS, S3 is applied to store the videos and connect to the transcribe service. Thirdly, transcribe is applied to transform the audio or video content into textual content and then make subtitles. The traditional process is expensive and require high degree of manual effort. The time require to make subtitle for a video can be multiple times of length of the video. Transcribe is a much more convenient choice that the whole process is automated and less time consuming. Transcribe access the source video/audio from a S3 storage therefore S3 is a must to connect the web and transcribe service.

2 Cost estimation

The cost estimation calculation is based on the AWS Pricing Calculator provided by Amazon in the link: <https://calculator.aws/#/> The development cost and fixed cost is negligible since that it is my own work and no cost applied.

2.1 Low scale (1-1,000 transactions/day)

2.1.2 EC2

From Real Numeracy (2012), the peak load/average load ratio for page below 1,000,000 views per month is 605 (select the more conservative number that close to the case). The peak transaction per second:

$$1000 \div 24 \div 3600 \times 605 = 7$$

According to Ernesto Marquez (2020), two t3.large instance is enough for EC2.

2.1.3 S3

Assume each transaction upload a video of 30 minutes video with size of around 200MB, total amount of storage required per month:

$$1000 \times 200 \times 30 = 6TB$$

2.1.4 Transcribe

According to Amazon pricing, the cost of a 30-minute video is \$0.72.

Total cost:

$$1000 \times 1.07 = \$720$$

2.1.5 Summary

My Estimate

Info

Add service

Add support

Add group

Action

Save and share

First 12 months total

2,793.96 USD

Total upfront

0.00 USD

Total monthly

232.83 USD

Services (2)

Amazon EC2

Region: US East (Ohio)

Edit

Action

Quick estimate

Operating system (Linux), Quantity (2), Storage for each EC2 instance (General Purpose SSD (gp2)), Storage amount (30 GB), Instance type (t3a.large)

Monthly:

74.77 USD

Amazon Simple Storage Service (S3)

Data Transfer

Edit

Action

S3 Standard storage (6 TB per month)

Monthly:

158.06 USD

Total cost is \$232.83(EC2 and S3) + \$720(transcribe) = \$952.83 USD per month.

2.2 Medium scale (1,000-1,000,000 transactions/day)

2.2.1 EC2

From Real Numeracy (2012), the peak load/average load ratio for page below 1,000,000,000 views per month is 3.7 (select the more conservative number that close to the case). The peak transaction per second:

$$1000000 \div 24 \div 3600 \times 3.7 = 43$$

According to Ernesto Marquez (2020), seven t3.large instances is enough for EC2.

2.2.2 S3

Assume each transaction upload a video of 30 minutes video with size of around 200MB, total amount of storage required per month:

$$1000000 \times 200 \times 30 = 6000TB$$

2.2.3 Transcribe

According to Amazon pricing, the cost of a 30-minute video is \$0.72.


Total cost:

$$1000000 \times 1.07 = \$720000$$

2.2.4 Summary

My Estimate Info

[Add service](#)[Add support](#)[Add group](#)[Action ▼](#)[Save and share](#)

**Contact Sales**
You might be able to further optimize your architecture and spend. To discuss your options, [contact sales](#).

First 12 months total

1,759,196.16 USD

Total upfront

0.00 USD

Total monthly

146,599.68 USD

Services (2)

Amazon Simple Storage Service (S3)

EditAction ▼

S3 Standard storage (6000 TB per month)

Monthly:146,338.00 USD

Data Transfer

Amazon EC2
Region: US East (Ohio)

EditAction ▼

Quick estimate

Operating system (Linux), Quantity (7), Storage for each EC2 instance (General Purpose SSD (gp2)), Storage amount (30 GB), Instance type (t3a.large)

Monthly:261.68 USD

The calculation is in link:

Total cost is \$146,600(EC2 and S3) + \$720,000(transcribe) ~ \$866,600 USD per month, which is less than 1000 times of the price of section 2.1.4.

However, special price is offered for all the service at this amount.

2.3 High scale (1,000,000+ transactions/day)

Assume 2,000,000 transactions/day

2.3.1 EC2

From Real Numeracy (2012), the peak load/average load ratio for page below 1,000,000,000 views per month is 3.7(select the more conservative number that close to the case). The peak transaction per second:

$$2000000 \div 24 \div 3600 \times 3.7 = 86$$

According to Ernesto Marquez (2020), 18 t3.medium instances is enough for EC2.

2.3.2 S3

Assume each transaction upload a video of 30 minutes video with size of around 200MB, total amount of storage required per month:

$$1000000 \times 200 \times 30 = 12000TB$$

2.3.3 Transcribe


According to Amazon pricing, the cost of a 30-minute video is \$0.72.

Total cost:

$$2000000 \times 1.07 = \$1440000$$

2.3.4 Summary

My Estimate [info](#) [Add service](#) [Add support](#) [Add group](#) [Action ▼](#) [Save and share](#)

 **Contact Sales**
You might be able to further optimize your architecture and spend. To discuss your options, [contact sales](#). [×](#)

First 12 months total	Total upfront	Total monthly
3,509,722.80 USD	0.00 USD	292,476.90 USD

Services (2)

Amazon Simple Storage Service (S3) [Edit](#) [Action ▼](#)

S3 Standard storage (12000 TB per month) Monthly: 292,112.80 USD

Data Transfer

Amazon EC2 [Edit](#) [Action ▼](#)
Region: US East (Ohio)

Quick estimate

Operating system (Linux), Quantity (18), Storage for each EC2 instance (General Purpose SSD (gp2)), Storage amount (30 GB), Instance type (t3a.medium) Monthly: 364.10 USD

Total cost is \$292,000(EC2 and S3) + \$1,440,000(transcribe) = \$1,732,000 USD per month.

Special price is also offered for all the service at this amount.

3 Installation manual

The EC2 machine adopted for this project is ubuntu with free tier. There are some tools or required for the installation:

1. PuTTY or SSH command to access the machine
2. WinSCP or scp command to transfer the files to the machine

3.1 Set up S3 Bucket

1. Create a S3 Bucket and open public access
2. Change the variable of flaskapp.py in the zip file on line 11 "itisprojectbucket" to your bucket name.

3.2 Create IAM role to allow EC2 accessing S3 and transcribe

1. Create an IAM role, select S3FullAccess and TranscribeFullAccess in policy









Create role 1 2 3 4

▼ Attach permissions policies

Choose one or more policies to attach to your new role.

Create policy ↺

Filter policies Q Search Showing 18 results

	Policy name	Used as
<input type="checkbox"/>	 AmazonEC2FullAccess	Permissions policy (1)
<input type="checkbox"/>	 AmazonElasticMapReduceforAutoScalingRole	Permissions policy (1)
<input type="checkbox"/>	 AmazonElasticMapReduceforEC2Role	Permissions policy (1)
<input type="checkbox"/>	 AmazonElasticMapReduceRole	Permissions policy (1)
<input checked="" type="checkbox"/>	 AmazonS3FullAccess	Permissions policy (2)
<input checked="" type="checkbox"/>	 AmazonTranscribeFullAccess	Permissions policy (2)
<input type="checkbox"/>	 AutoScalingServiceRolePolicy	Permissions policy (1)
<input type="checkbox"/>	 AWSCloud9ServiceRolePolicy	Permissions policy (1)

2. Name the role whatever your like and record it.

3.3 Create EC2 Ubuntu instance

1. Choose Ubuntu Server 18.04 LTS (HVM) for AMI (version may vary from time to time).
2. Open the following ports in security setting.

Type i	Protocol i	Port Range i	Source i	Description i
HTTP	TCP	80	0.0.0.0/0	
HTTP	TCP	80	::/0	
SSH	TCP	22	0.0.0.0/0	for ssh
Custom TCP Rule	TCP	5000	0.0.0.0/0	
HTTPS	TCP	443	0.0.0.0/0	

3. Launch the instance.
4. Attach the IAM role created in section 3.2 to the instance to access S3 and transcribe through EC2.

3.4 Set up the server

1. Connect to EC2 instance through SSH/PuTTY

Change python3 to default python version due to compatibility issues

```
sudo apt-get update
```

```
update-alternatives --install /usr/bin/python python /usr/bin/python3 10
```

2. Install apache webserver and mod_wsgi

```
sudo apt-get install apache2
```

```
sudo apt-get install libapache2-mod-wsgi-py3
```

3. Install pip3 and the following packages for python: flask, flask-bootstrap, boto3, requests and arrow.

```
sudo apt-get -y install python3-pip
```

```
sudo pip3 install flask
```

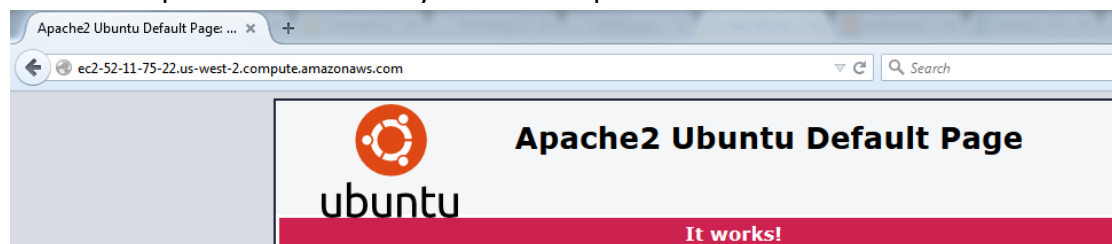
```
sudo pip3 install flask-bootstrap
```

```
sudo pip3 install boto3
```

```
sudo pip3 install requests
```

```
sudo pip3 install arrow
```

4. Check the apache server is ok by browse the public DNS of instance










5. Create directory for flask app

```
mkdir ~/flaskapp
```

```
sudo ln -sT ~/flaskapp /var/www/html/flaskapp
```

6. Copy the files in zip to /home/ubuntu/flaskapp directory through WinSCP

/home/ubuntu/flaskapp/				
Name	Size	Changed	Rights	Owner
		1/06/2020 10:28:20 AM	rw-r--r--	ubuntu
 static		1/06/2020 10:29:55 AM	rw-r--r--	ubuntu
 templates		1/06/2020 10:30:01 AM	rw-r--r--	ubuntu
 filters.py	1 KB	31/05/2020 4:25:30 AM	rw-r--r--	ubuntu
 flaskapp.py	4 KB	31/05/2020 3:38:36 AM	rw-r--r--	ubuntu
 flaskapp.wsgi	1 KB	27/05/2020 12:50:12 PM	rw-r--r--	ubuntu
 transcribe.py	2 KB	31/05/2020 4:25:29 AM	rw-r--r--	ubuntu

7. Enable mod_wsgi.

The apache server displays html pages by default but to serve dynamic content

from a Flask app we'll have to make a few changes. In the apache configuration file located at /etc/apache2/sites-enabled/000-default.conf, add the following block just after the DocumentRoot /var/www/html line:

```
File Edit Options Buffers Tools Conf Help
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

    WSGIDaemonProcess flaskapp threads=5
    WSGIScriptAlias / /var/www/html/flaskapp/flaskapp.wsgi

    <Directory flaskapp>
        WSGIProcessGroup flaskapp
        WSGIApplicationGroup %{GLOBAL}
        Order deny,allow
        Allow from all
    </Directory>

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
- UU-:----Fi 000-default.conf Top Li (Conf[Space]) -----
```

```
WSGIDaemonProcess flaskapp threads=5
```

```
WSGIScriptAlias / /var/www/html/flaskapp/flaskapp.wsgi
```

```
<Directory flaskapp>
```

```
    WSGIProcessGroup flaskapp
```

```
    WSGIApplicationGroup %{GLOBAL}
```

```
    Order deny,allow
```

```
    Allow from all
```

```
</Directory>
```

8. Change the config of apache to set 5000 as default port and restart apache.

The installation is completed!

4 Reference:

Real Numeracy, 2012. Estimating peak load from average load, url:

<https://realnumeracy.wordpress.com/2012/01/28/estimating-peak-load-from-average-load/>

Ernesto Marquez, 2020. How to find an optimal EC2 configuration in 5 steps (with actual performance tests and results), url:

<https://www.concurrencylabs.com/blog/5-steps-for-finding-optimal-ec2-infrastructure/>

Amazon Transcribe Pricing, url: <https://aws.amazon.com/transcribe/pricing/>

Deploy A Flask App On AWS EC2, Joaquin Guardado, 2020. url:

<https://www.codementor.io/@jqn/deploy-a-flask-app-on-aws-ec2-13hp1ilqy2>

Flask with Amazon S3, KishStats, 2018. url:

<https://kishstats.com/python/2018/03/15/flask-amazon-s3-part-1.html>

AWS Transcribe Developer Guide, AWS. url:

<https://docs.aws.amazon.com/transcribe/latest/dg/getting-started-python.html>

Convert Amazon Transcribe JSON to SRT, Yash. url: <https://www.yash.info/aws-srt-creator.htm>