

Name: Jaime Valencia

Program: CLOUD DEVELOPER

Project: #2 Udagram Image Filtering Microservice

Github repo: <https://github.com/vaduinc/cloud-developer>

Environments

Elastic Beanstalk > Environments

All environments **API application** **Image Process Application**

Filter results matching the displayed values

Environment name	Health	Application name	Date created	Last modified	URL	Running versions	Platform	Platform state	Tier name
udagramjvapi	Ok	udagramjvapidv	2020-04-10 01:21:24 UTC-0400	2020-04-12 22:18:43 UTC-0400	udagramjvapi.us-east-1.elasticbeanstalk.com	app-200412_145413	Node.js running on 64bit Amazon Linux	Supported	WebServer
udagramjvimagedev-dev	Ok	udagramjvimagedev	2020-04-12 14:35:59 UTC-0400	2020-04-12 19:01:55 UTC-0400	udagramjvimagedev-dev.us-east-1.elasticbeanstalk.com	app-200412_143555	Node.js running on 64bit Amazon Linux	Supported	WebServer

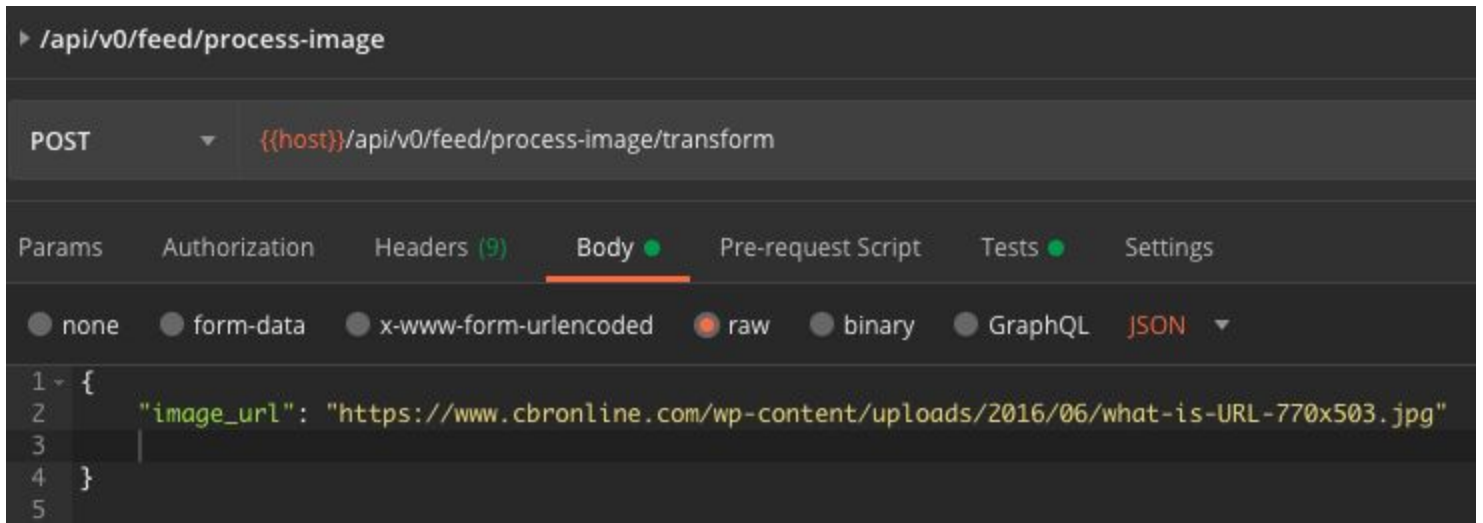
Environment variable 'IMG_API_HOST' added to 'udagramjvapidv' to communicate (send request) to the image process server.

Environment properties

The following properties are passed in the application as environment properties. [Learn more](#)

Name	Value
AWS_MEDIA_BUCKET	udagramjvs3dev
AWS_PROFILE	DEPLOYED
AWS_REGION	us-east-1
IMG_API_HOST	http://udagramjvimagedev-dev
JWT_SECRET	
POSTGRES_DB	udagramjvdbdev
POSTGRES_HOST	udagramjvdbdev.cah015zgwcj
POSTGRES_PASSWORD	
POSTGRES_USERNAME	

A *POST* end-point was added (`/api/v0/feed/process-image/transform`) to receive the request from clients (front-end) to forward the request to the process image application/server.



Elastic Beanstalk (Image process application) ‘udagramjvimagedev’

URL: <http://udagramjvimagedev-dev.us-east-1.elasticbeanstalk.com/>

Elastic Beanstalk > Applications > udagramjvimagedev

Application 'udagramjvimagedev' environments Create a new environment

Filter results matching the display values

Environment name	Health	Date created	Last modified	URL	Running versions	Platform	Platform state	Tier name
udagramjvimagedev-dev	Ok	2020-04-12 14:35:59 UTC-0400	2020-04-12 19:01:55 UTC-0400	udagramjvimagedev-dev.us-east-1.elasticbeanstalk.com	app-200412_143555	Node.js running on 64bit Amazon Linux	Supported	WebServer

udagramjvimagedev-dev[udagramjvimagedev-dev.us-east-1.elasticbeanstalk.com \(e-pzdbp9trtd\)](#)Application name: **udagramjvimagedev**

Refresh

Environment actions ▼

Health

Ok

Causes

Running version

app-200412_143555

Upload and deploy

PlatformNode.js running on 64bit Amazon
Linux/4.14.1

Change

The S3 bucket is configured to support static website hosting.

Buckets (3)

Copy ARN

Empty

Delete

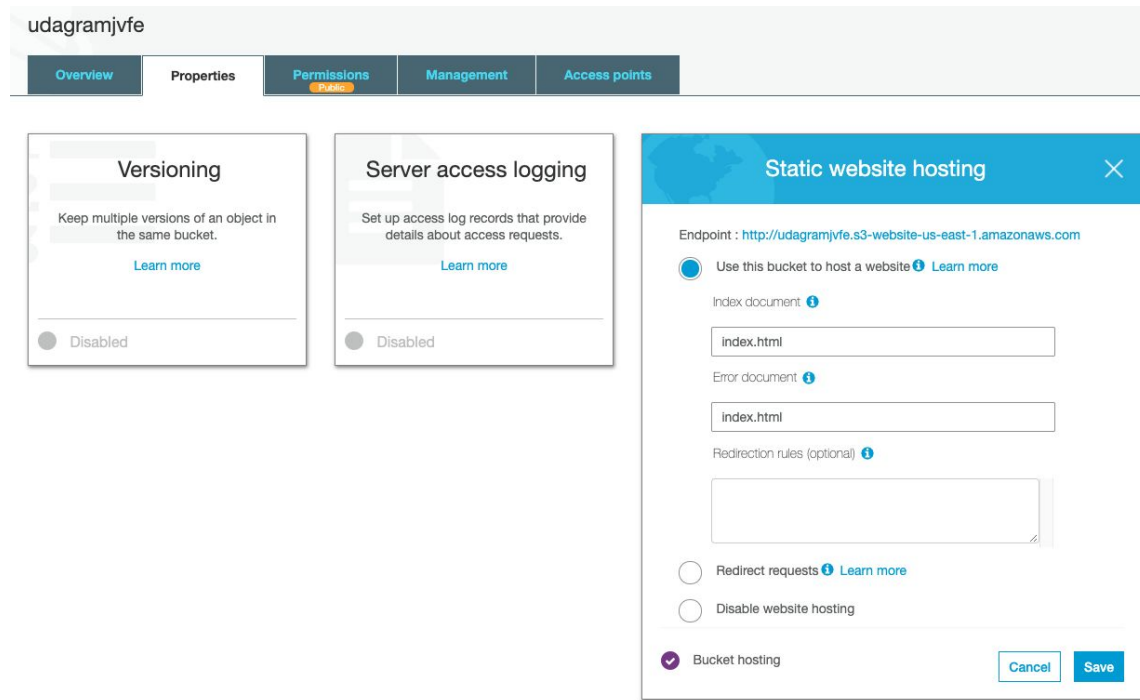
Create bucket

Find bucket by name

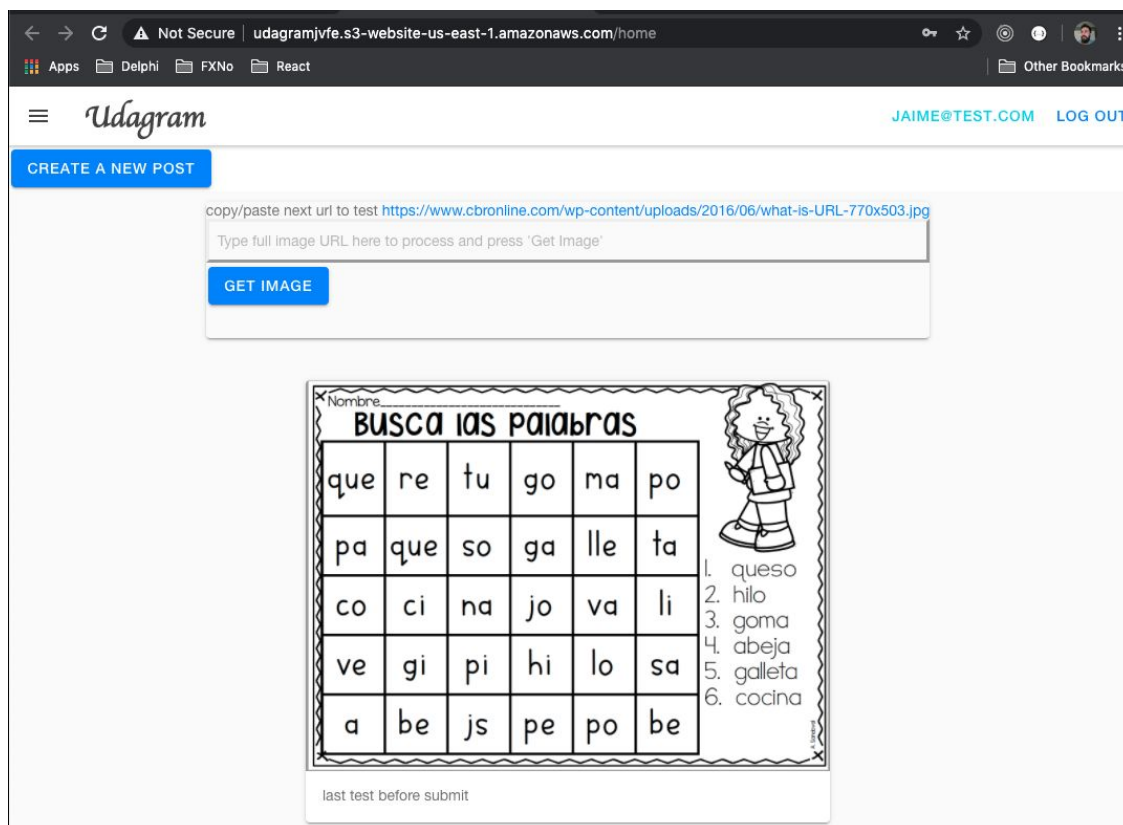
**Ionic Front-End
application****Bucket to
store images**

	Name	Region	Access	Bucket created
<input type="radio"/>	udagramjvfe	US East (N. Virginia) us-east-1	Public	2020-04-11T21:05:59.000Z
<input type="radio"/>	elasticbeanstalk-us-east-1-73400343731	US East (N. Virginia) us-east-1	Objects can be public	2020-04-10T04:52:24.000Z
<input type="radio"/>	udagramjvs3dev	US East (N. Virginia) us-east-1	Not Public	2020-04-09T16:48:00.000Z

Front-End



FRONT-END URL: <http://udagramjvfe.s3-website-us-east-1.amazonaws.com/home>



Test image processing front-end

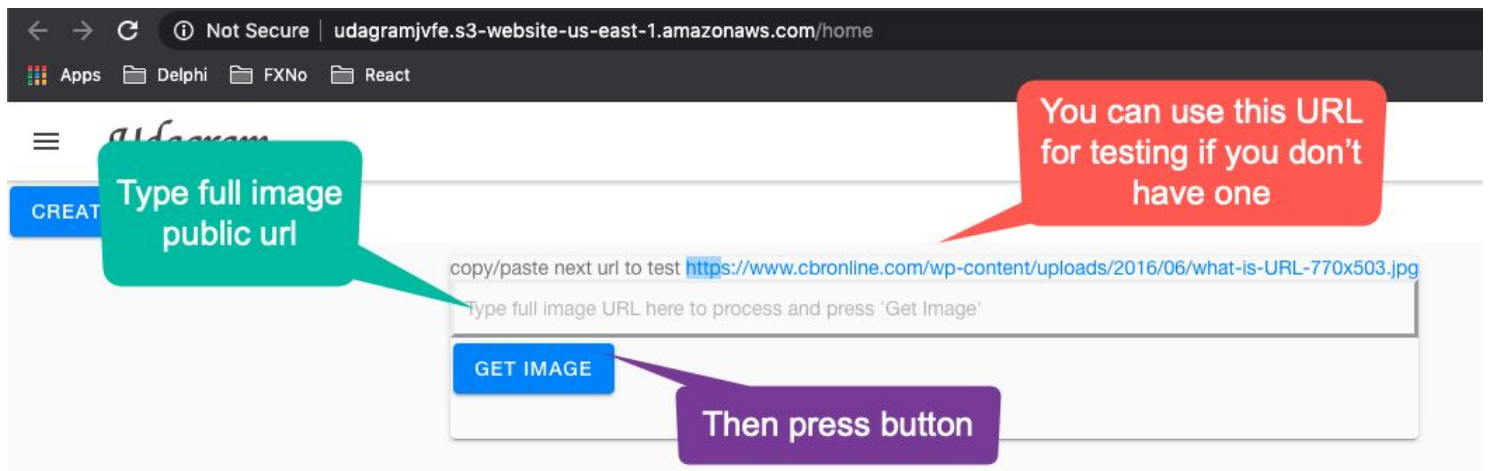
Request is made to the API application (udagramjvapidew) and there the system makes a request to the Image Processing server (udagramjvimagedev).

This was supposed to be used with each of the images from the DB. However, I always got back an error and unfortunately, I wasn't able to make work with those URLs.

Here is the error. I know the S3 returns XML type exceptions.

```
(node:25043) UnhandledPromiseRejectionWarning: Error: Unsupported MIME type: application/xml
    at Jimp.throwError (/Users/jaimevalencia/Documents/udacity/cloudDeveloper/project2/cloud-developer/course-02/project/image-filter-starter-code/node_modules/@jimp/utis/src/index.js:15:13)
    at Jimp.call (/Users/jaimevalencia/Documents/udacity/cloudDeveloper/project2/cloud-developer/course-02/project/image-filter-starter-code/node_modules/@jimp/core/src/utis/image-bitmap.js:84:25)
    at Jimp.call [as parseBitmap] (/Users/jaimevalencia/Documents/udacity/cloudDeveloper/project2/cloud-developer/course-02/project/image-filter-starter-code/node_modules/@jimp/core/src/index.js:400:17)
    at parseBitmap (/Users/jaimevalencia/Documents/udacity/cloudDeveloper/project2/cloud-developer/course-02/project/image-filter-starter-code/node_modules/@jimp/core/src/index.js:344:14)
    at cb (/Users/jaimevalencia/Documents/udacity/cloudDeveloper/project2/cloud-developer/course-02/project/image-filter-starter-code/node_modules/@jimp/core/src/index.js:73:14)
    at cb (/Users/jaimevalencia/Documents/udacity/cloudDeveloper/project2/cloud-developer/course-02/project/image-filter-starter-code/node_modules/@jimp/core/src/request.js:47:9)
    at IncomingMessage.<anonymous> (/Users/jaimevalencia/Documents/udacity/cloudDeveloper/project2/cloud-developer/course-02/project/image-filter-starter-code/node_modules/phn/lib/phn.compiled.js:1:2038)
    at IncomingMessage.emit (events.js:333:22)
    at endReadableNT (_stream_readable.js:1220:12)
    at processTicksAndRejections (internal/process/task_queues.js:84:21)
```

Following is the explanation on how to use the front-end to transform an image.

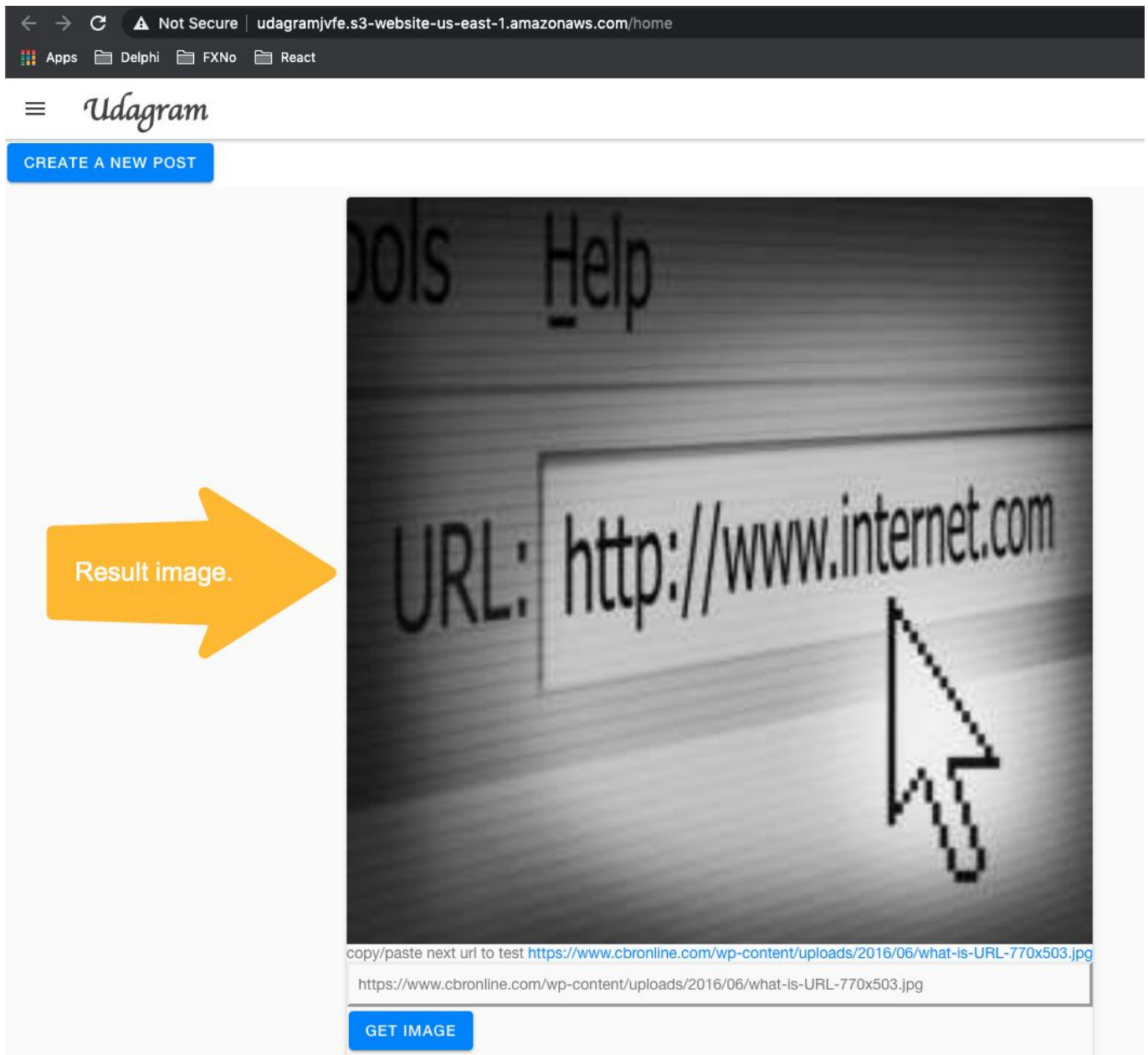


Following there is an example of taking the following image from the Internet and processing through the application and rendering the result.

Original image URL <https://www.cbronline.com/wp-content/uploads/2016/06/what-is-URL-770x503.jpg>



Following is the resulting image after processing.



Postmand Collections

course-02/exercises/udacity-c2-restapi/udacity-c2-restapi.postman_collection.json

course-02/project/image-filter-starter-code/JaimeValencia-project2.postman_collection.json