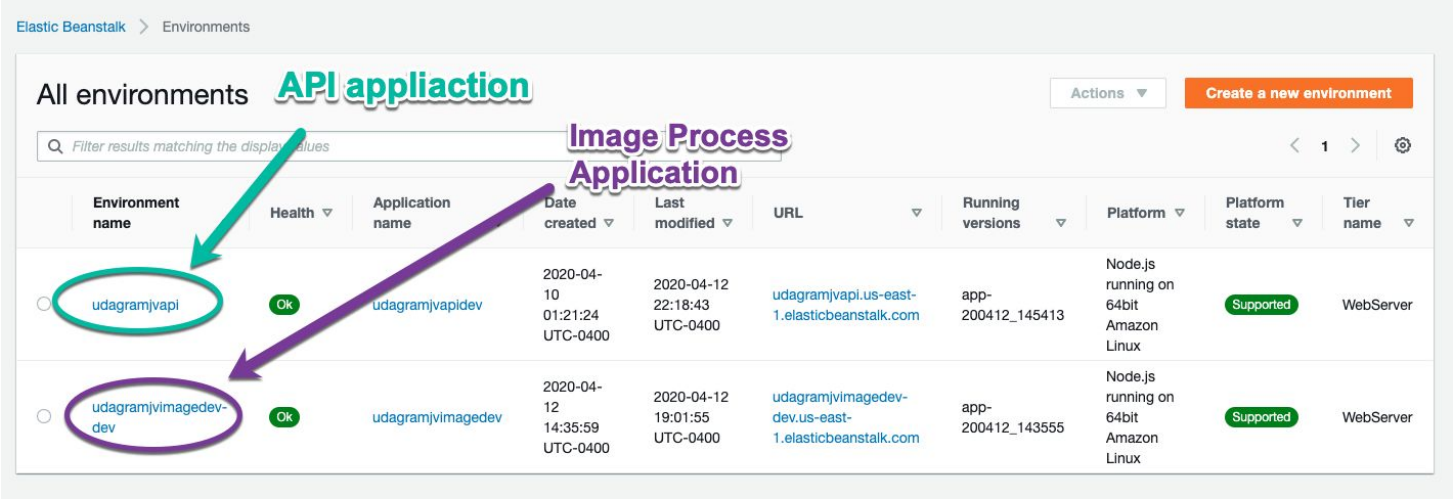
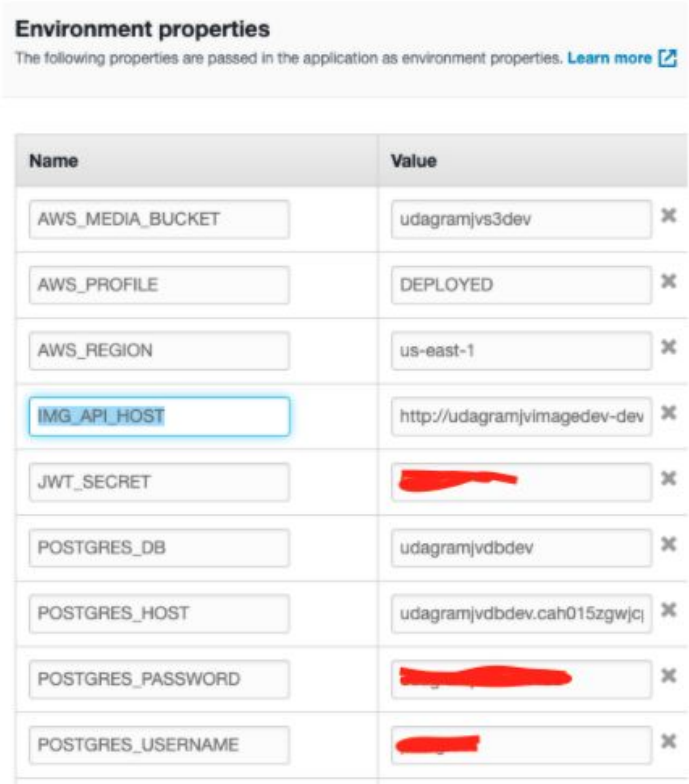
Name: Jaime Valencia
Program: CLOUD DEVELOPER
Project: #2 **Udagram Image Filtering Microservice**
Github repo: https://github.com/vaduinc/cloud-developer

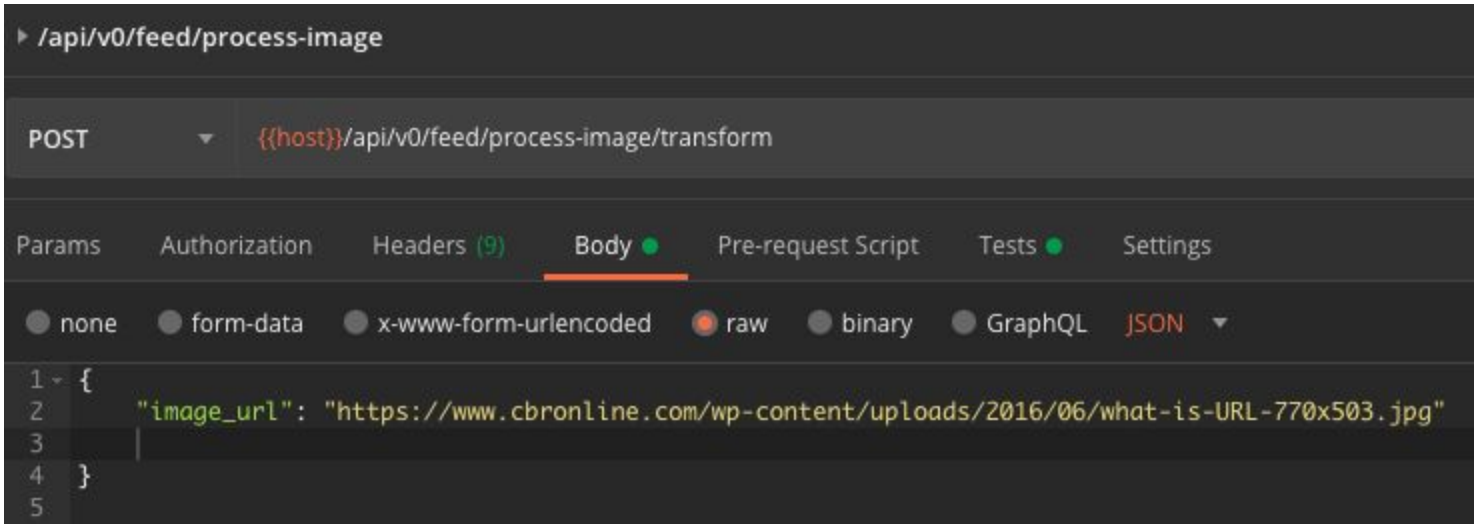
Environments



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

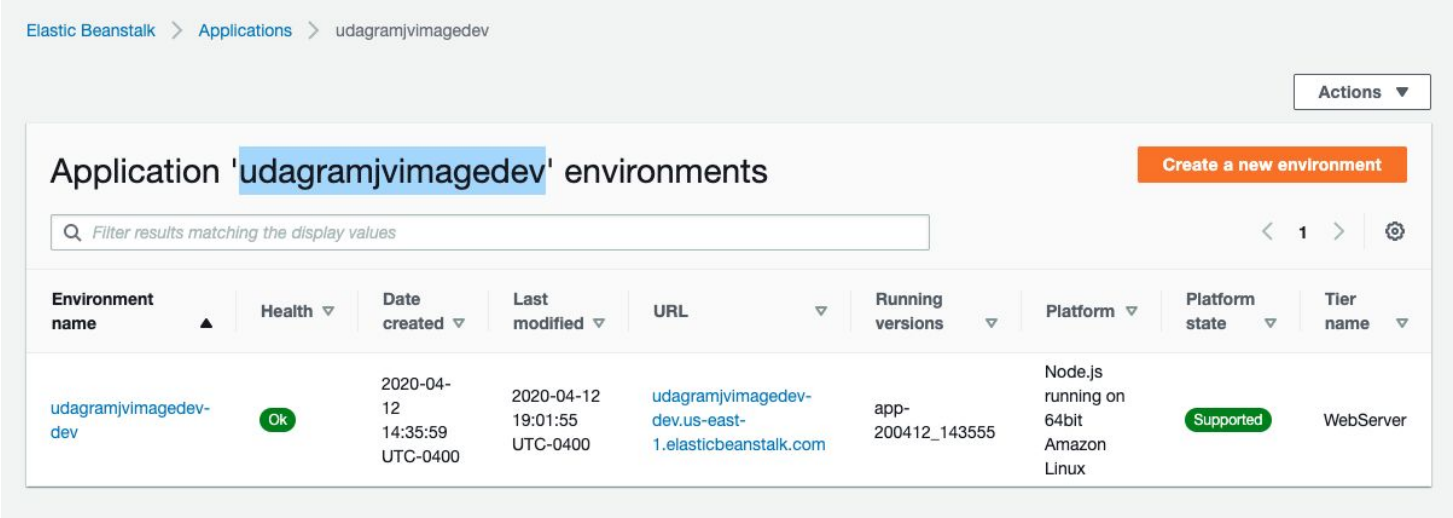


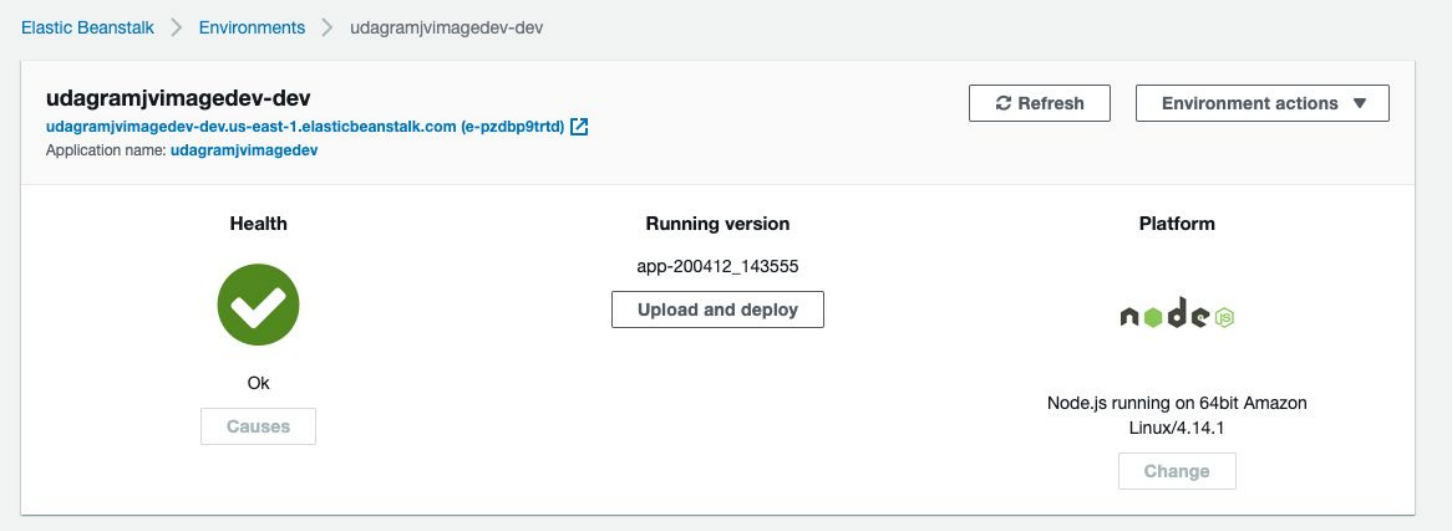
A *POST* end-point was added (`/api/v0/feed/process-image/transform`) to receive the request from clients (front-end) to forward the request to the process image application/server.



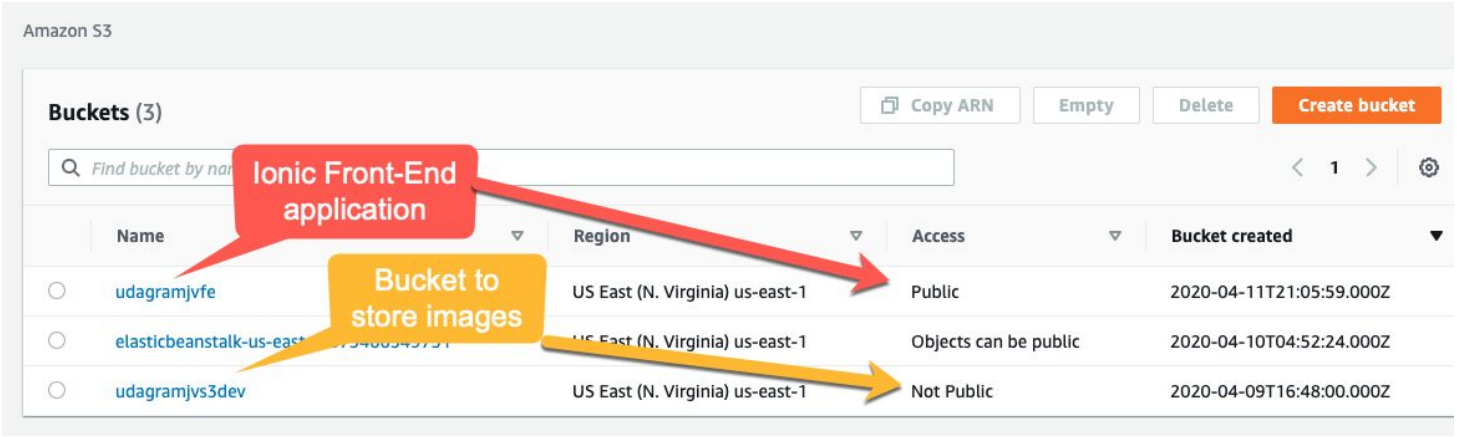
Elastic Beanstalk (Image process application) 'udagramjvimagedev'

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

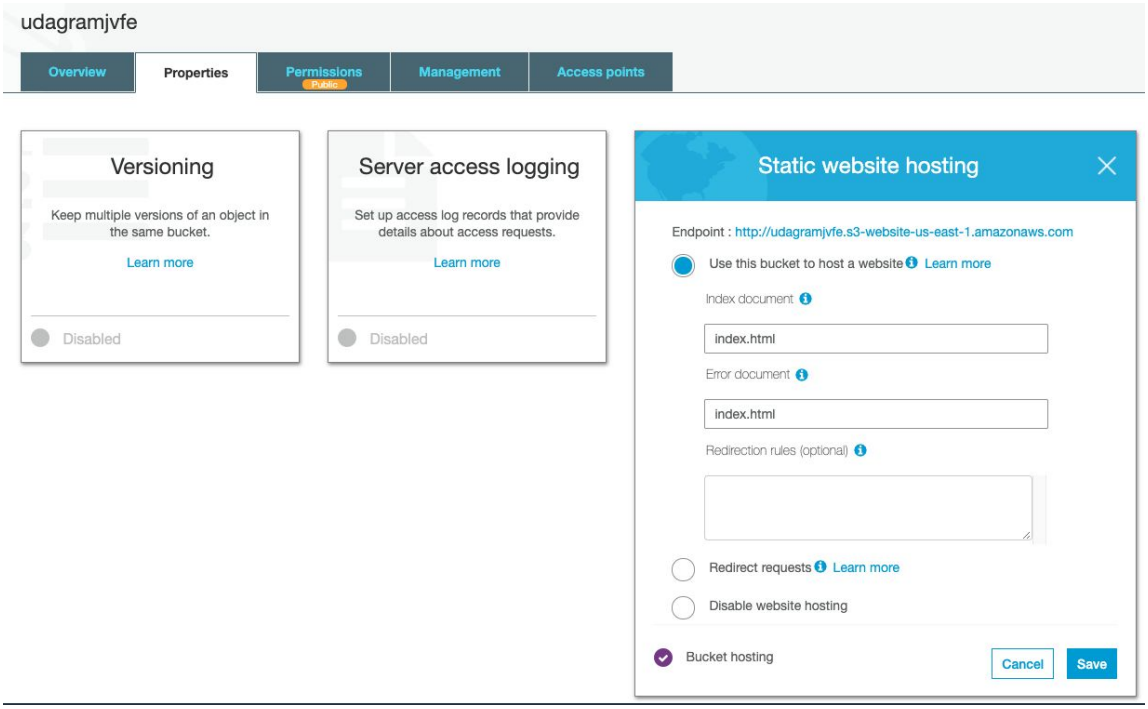




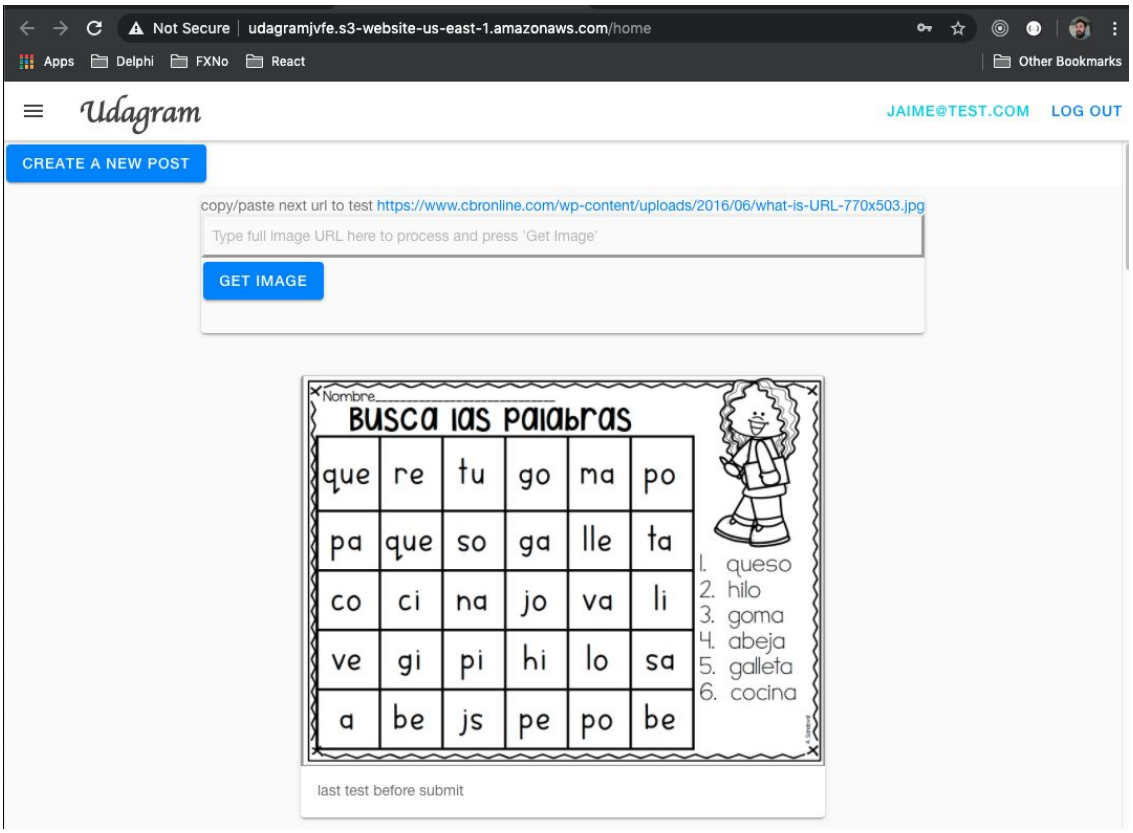
The S3 bucket is configured to support static website hosting.



Front-End



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



Test image processing front-end

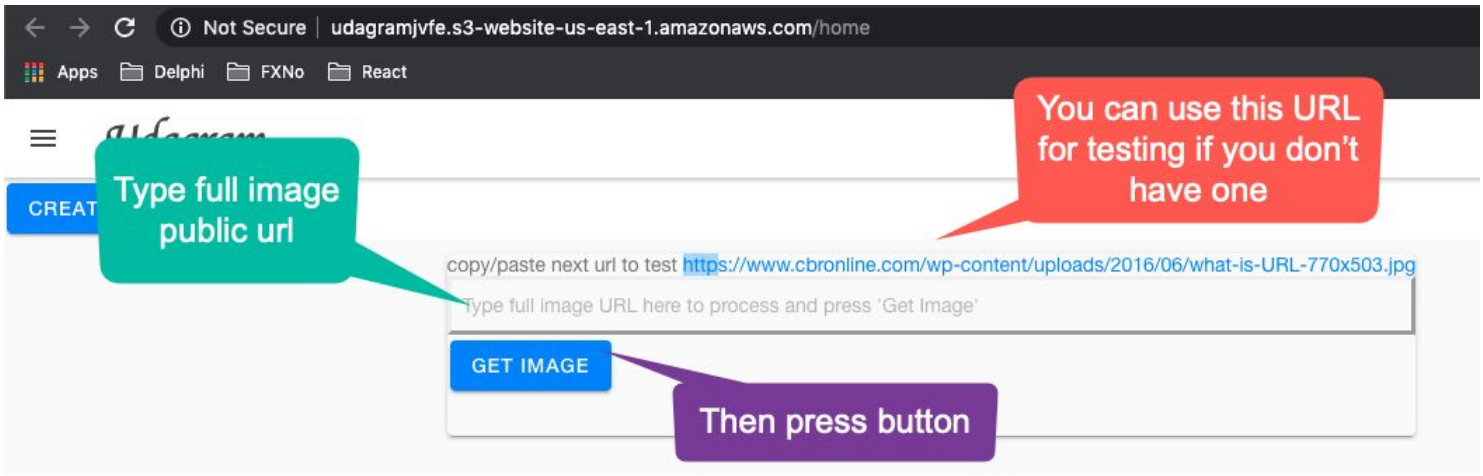
Request is made to the API application (udagramjvapidev) and there the system makes a request to the Image Processing server (udagramjvimagedev).

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.

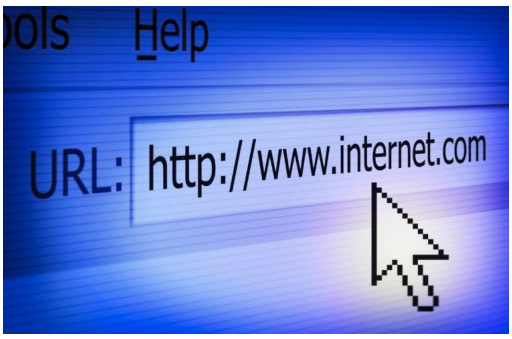
```
(node:25043) UnhandledPromiseRejectionWarning: Error: Unsupported MIME type: application/xml
    at Jimp.throwError (/Users/jaimevalencia/Documents/udacity/cloudDeveloper/project2/cloud-developer/course-02/project/image-filter-starter-code/node_modules/@jimp/utils/src/index.js:15:13)
    at Jimp.call (/Users/jaimevalencia/Documents/udacity/cloudDeveloper/project2/cloud-developer/course-02/project/image-filter-starter-code/node_modules/@jimp/core/src/utils/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/phin/lib/phin.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.

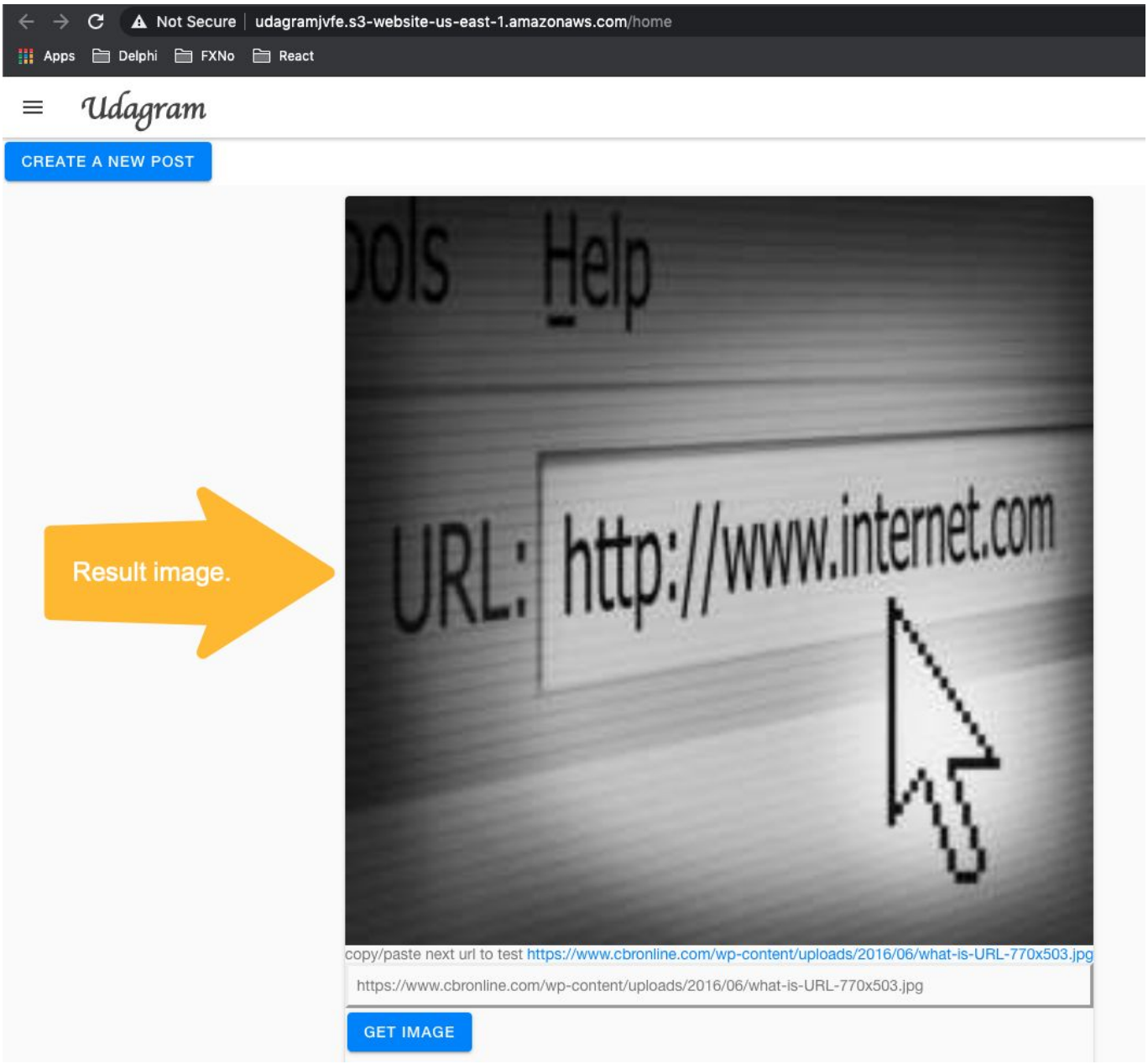


Following there is an example of taking the following image from the Internet and processing through the application and rendering the result.

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