

Image Filter Code Dev

Engineering Process and Quality

1-The project demonstrates an understanding of a good cloud git process,

I have included two links for both branches.

Master branch

<https://github.com/Mostafaelfekky/cloud-developer/tree/master/course-02/project/image-filter-starter-code>

Dev branch

<https://github.com/Mostafaelfekky/cloud-developer/tree/Dev/course-02/project/image-filter-starter-code>

2-The project demonstrates an ability to use typescript and Nodejs.

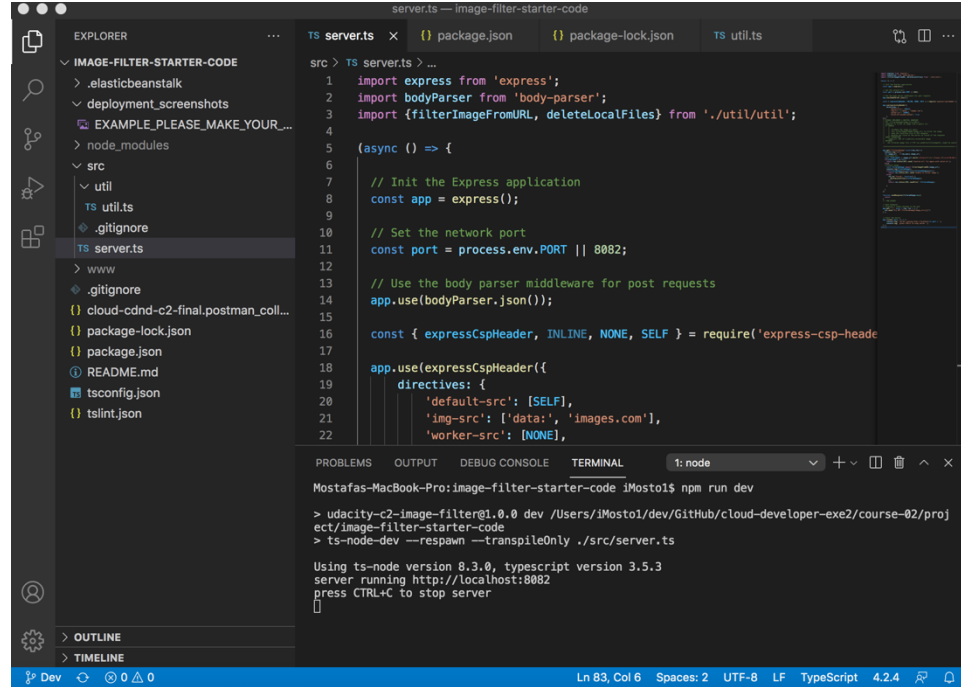
The code changes were clear with the functions and variables names, I have included the link for the code.

<https://github.com/Mostafaelfekky/cloud-developer/blob/Dev/course-02/project/image-filter-starter-code/src/server.ts>

Development Server

1-The project demonstrates the ability to develop using the NodeJS framework,

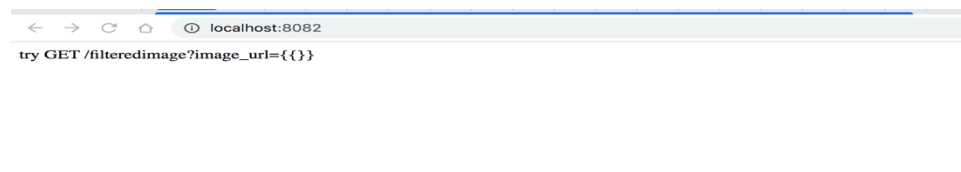
-It was successfully running locally on port 8082.



The screenshot shows the Visual Studio Code interface with a project named 'image-filter-starter-code'. The Explorer panel on the left shows the file structure, including 'src' and 'util' folders. The main editor displays the 'server.ts' file with the following code:

```
src > TS server.ts > ...
1 import express from 'express';
2 import bodyParser from 'body-parser';
3 import {filterImageFromURL, deleteLocalFiles} from './util/util';
4
5 (async () => {
6
7     // Init the Express application
8     const app = express();
9
10    // Set the network port
11    const port = process.env.PORT || 8082;
12
13    // Use the body parser middleware for post requests
14    app.use(bodyParser.json());
15
16    const { expressCspHeader, INLINE, NONE, SELF } = require('express-csp-header');
17
18    app.use(expressCspHeader({
19      directives: {
20        'default-src': [SELF],
21        'img-src': ['data:', 'images.com'],
22        'worker-src': [NONE],
```

The TERMINAL panel at the bottom shows the command 'npm run dev' being executed, resulting in the server running on http://localhost:8082.



2-The project demonstrates an understanding of RESTFUL design.

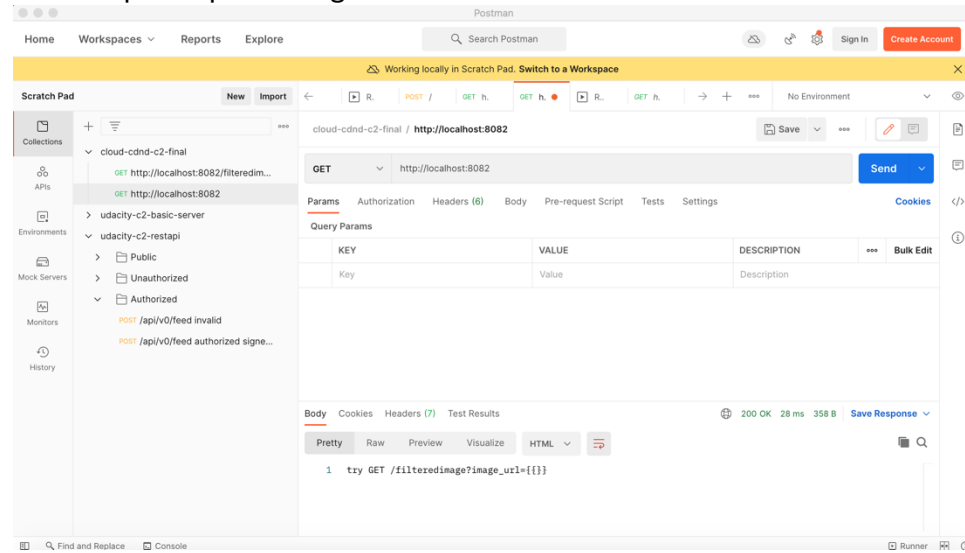
-We were able to run the image and have a successful output after filtering the image.



-The output of postman gave us 200 HTTP status

3-The project demonstrates an understanding of HTTP status codes.

-The output of postman gave us Code 200 HTTP status as observed



Elastic Beanstalk Deployment

1-The project demonstrates the ability to create functional cloud deployments.

Providing the EB link:

http://imagefilter-dev.eba-4au6xntm.us-east-1.elasticbeanstalk.com/filteredimage?image_url=https://upload.wikimedia.org/wikipedia/commons/b/bd/Golden_tabby_and_white_kitten_n01.jpgg

A screenshot of the output



2-The project demonstrates an understanding of AWS Elastic Beanstalk's CLI and Console Dashboard.

Providing you screenshots of the workflow of the deployment.

-Running `eb init`

```
Mostafas-MacBook-Pro:image-filter-starter-code iMosto1$ eb init

Select a default region
1) us-east-1 : US East (N. Virginia)
2) us-west-1 : US West (N. California)
3) us-west-2 : US West (Oregon)
4) eu-west-1 : EU (Ireland)
5) eu-central-1 : EU (Frankfurt)
6) ap-south-1 : Asia Pacific (Mumbai)
7) ap-southeast-1 : Asia Pacific (Singapore)
8) ap-southeast-2 : Asia Pacific (Sydney)
9) ap-northeast-1 : Asia Pacific (Tokyo)
10) ap-northeast-2 : Asia Pacific (Seoul)
11) sa-east-1 : South America (Sao Paulo)
12) cn-north-1 : China (Beijing)
13) cn-northwest-1 : China (Ningxia)
14) us-east-2 : US East (Ohio)
15) ca-central-1 : Canada (Central)
16) eu-west-2 : EU (London)
17) eu-west-3 : EU (Paris)
18) eu-north-1 : EU (Stockholm)
19) eu-south-1 : EU (Milano)
20) ap-east-1 : Asia Pacific (Hong Kong)
21) me-south-1 : Middle East (Bahrain)
22) af-south-1 : Africa (Cape Town)
(default is 3): 1

Select an application to use
1) image-filter-starter-code
2) Udiagram-Most-dev
3) [ Create new Application ]
(default is 1): 1

It appears you are using Node.js. Is this correct?
(Y/n): Y
Select a platform branch.
1) Node.js 14 running on 64bit Amazon Linux 2
2) Node.js 12 running on 64bit Amazon Linux 2
3) Node.js running on 64bit Amazon Linux
4) Node.js 10 running on 64bit Amazon Linux 2 (Deprecated)
(default is 1): 2

Cannot setup CodeCommit because there is no Source Control setup, continuing with initialization
```

-Running `eb create`

```
(default is 1): 2

Cannot setup CodeCommit because there is no Source Control setup, continuing with initialization
Do you want to set up SSH for your instances?
(Y/n): Y

Select a keypair.
1) UDAcity
2) vockey
3) [ Create new KeyPair ]
(default is 2): 2

Mostafas-MacBook-Pro:image-filter-starter-code iMosto1$ eb create
Enter Environment Name
(default is image-filter-starter-code-dev):
Enter DNS CNAME prefix
(default is image-filter-starter-code-dev22222222):

Select a load balancer type
1) classic
2) application
3) network
(default is 2):

Would you like to enable Spot Fleet requests for this environment? (y/N):
Creating application version archive "app-210529_133625".
Uploading: [#####] 100% Done...
Environment details for: image-filter-starter-code-dev
  Application name: image-filter-starter-code
  Region: us-east-1
  Deployed Version: app-210529_133625
  Environment ID: e-quz2smzt29
  Platform: arn:aws:elasticbeanstalk:us-east-1::platform/Node.js 12 running on 64bit Amazon Linux 2/5.3.2
  Tier: WebServer-Standard-1.0
  CNAME: image-filter-starter-code-dev22222222.us-east-1.elasticbeanstalk.com
  Updated: 2021-05-29 11:36:52.095000+00:00
Printing Status:
2021-05-29 11:36:50 INFO createEnvironment is starting.
2021-05-29 11:36:52 INFO Using elasticbeanstalk-us-east-1-103987518368 as Amazon S3 storage
  bucket for environment data.
2021-05-29 11:37:14 INFO Created target group named: arn:aws:elasticloadbalancing:us-east-1
:103987518368:targetgroup/awseb-AWSEB-1PZL6Q10GKWQR/7efd03b47bb21567
2021-05-29 11:37:14 INFO Created security group named: sg-0a0d56e00283dd4f2
2021-05-29 11:37:30 INFO Created security group named: awseb-e-quz2smzt29-stack-AWSEBSecuri
```

And finally, it was deployed successful.

The screenshot shows the AWS Elastic Beanstalk console. The left sidebar lists 'Elastic Beanstalk' with options for 'Environments', 'Applications', and 'Change history'. Under 'Environments', 'IMAGE-FILTER' is expanded, showing 'Application versions' and 'Saved configurations'. The 'Imagefilter-DEV' environment is selected, showing its details: 'Imagefilter-DEV' (Application name: IMAGE-FILTER), 'Health' (Ok), 'Running version' (image-filter-source), and 'Platform' (Node.js 12 running on 64bit Amazon Linux 2/5.3.2). The 'Recent events' section at the bottom shows a list of events with a 'Show all' button.

Name of the Student:

Mostafa Zakaria Elfekky