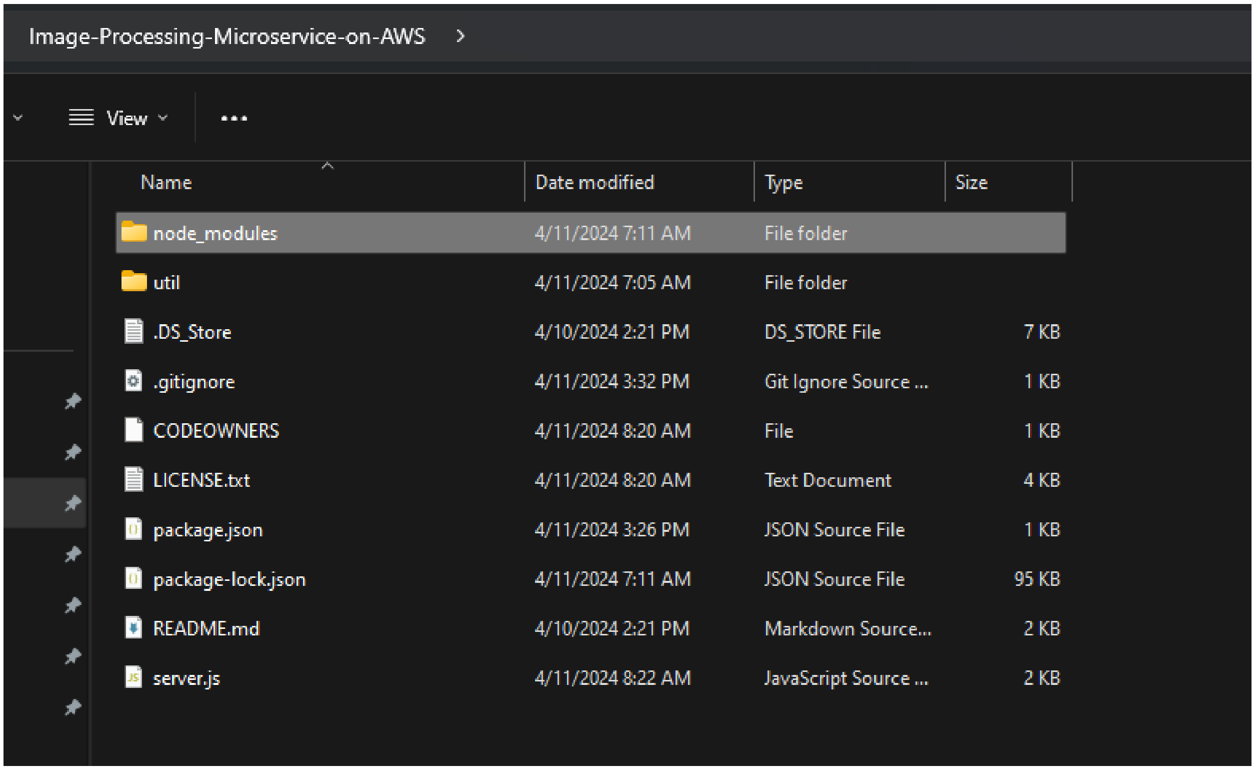
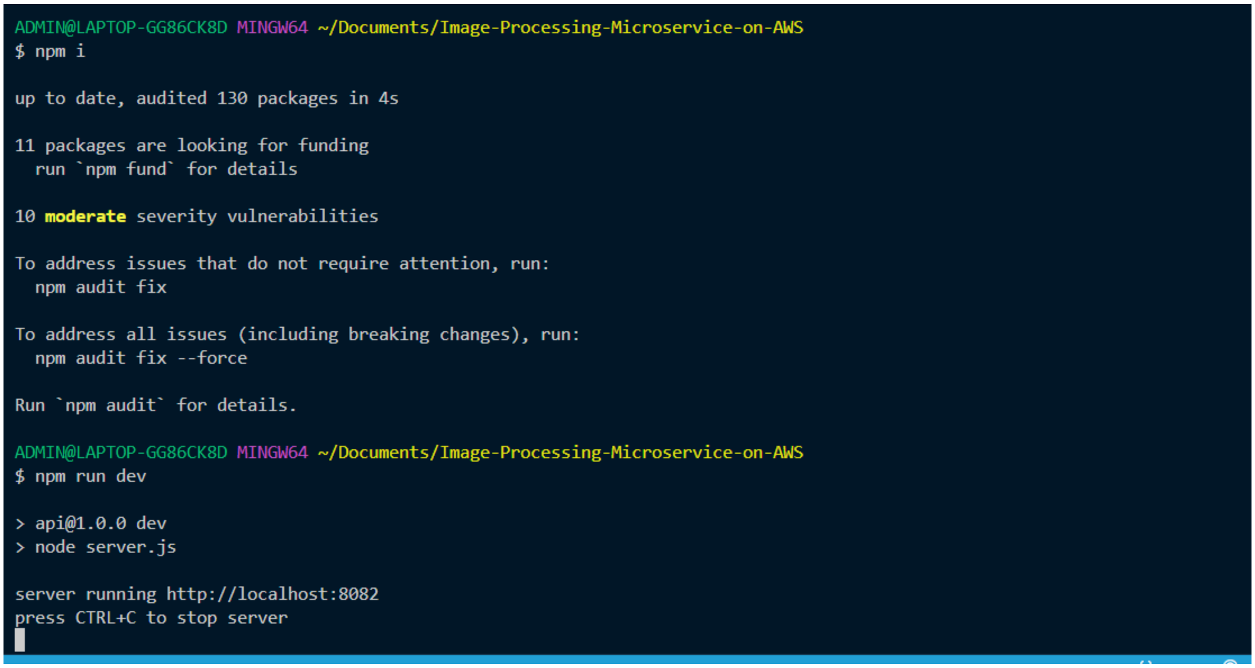
Project: Image Processing Microservice on AWS

- Step 1: Download source code from github Image Filter Starter Code and reorganize the project folder with the following structure (my project named Image-Processing-Microservice-on-AWS):



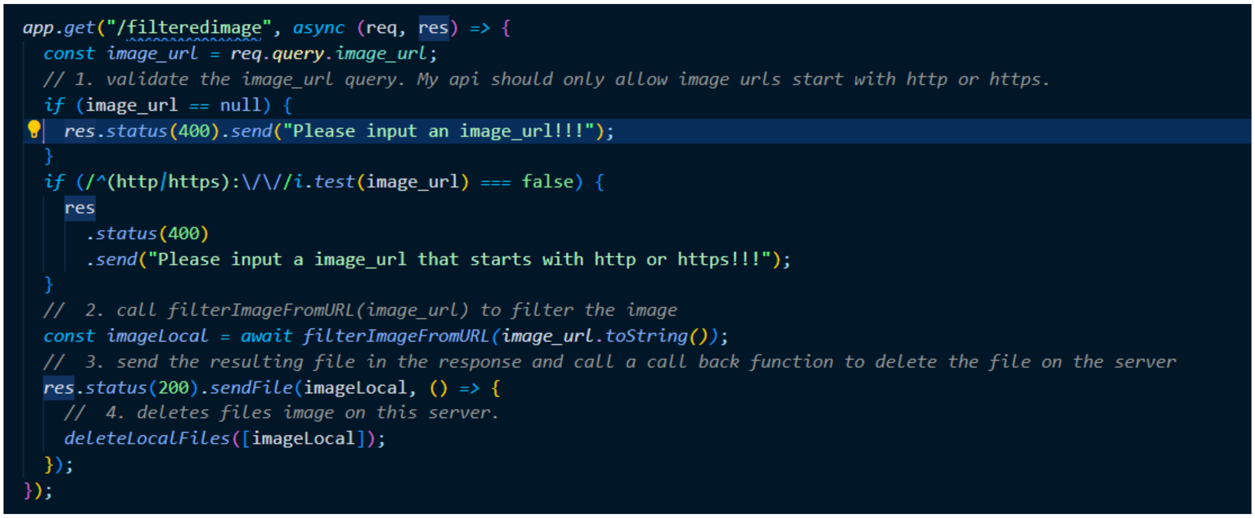
Name	Date modified	Type	Size
node_modules	4/11/2024 7:11 AM	File folder	
util	4/11/2024 7:05 AM	File folder	
.DS_Store	4/10/2024 2:21 PM	DS_STORE File	7 KB
.gitignore	4/11/2024 3:32 PM	Git Ignore Source ...	1 KB
CODEOWNERS	4/11/2024 8:20 AM	File	1 KB
LICENSE.txt	4/11/2024 8:20 AM	Text Document	4 KB
package.json	4/11/2024 3:26 PM	JSON Source File	1 KB
package-lock.json	4/11/2024 7:11 AM	JSON Source File	95 KB
README.md	4/10/2024 2:21 PM	Markdown Source...	2 KB
server.js	4/11/2024 8:22 AM	JavaScript Source ...	2 KB

- Step 2: Set up environment:

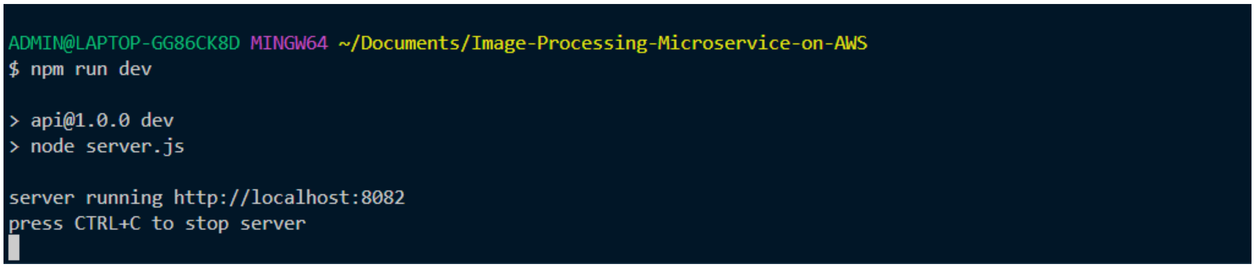


```
ADMIN@LAPTOP-GG86CK8D MINGW64 ~/Documents/Image-Processing-Microservice-on-AWS
$ npm i

up to date, audited 130 packages in 4s

11 packages are looking for funding
  run `npm fund` for details

10 moderate severity vulnerabilities

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.

ADMIN@LAPTOP-GG86CK8D MINGW64 ~/Documents/Image-Processing-Microservice-on-AWS
$ npm run dev

> api@1.0.0 dev
> node server.js

server running http://localhost:8082
press CTRL+C to stop server
```

- Step 3: Create a new endpoint in the server.js file and test in postman:

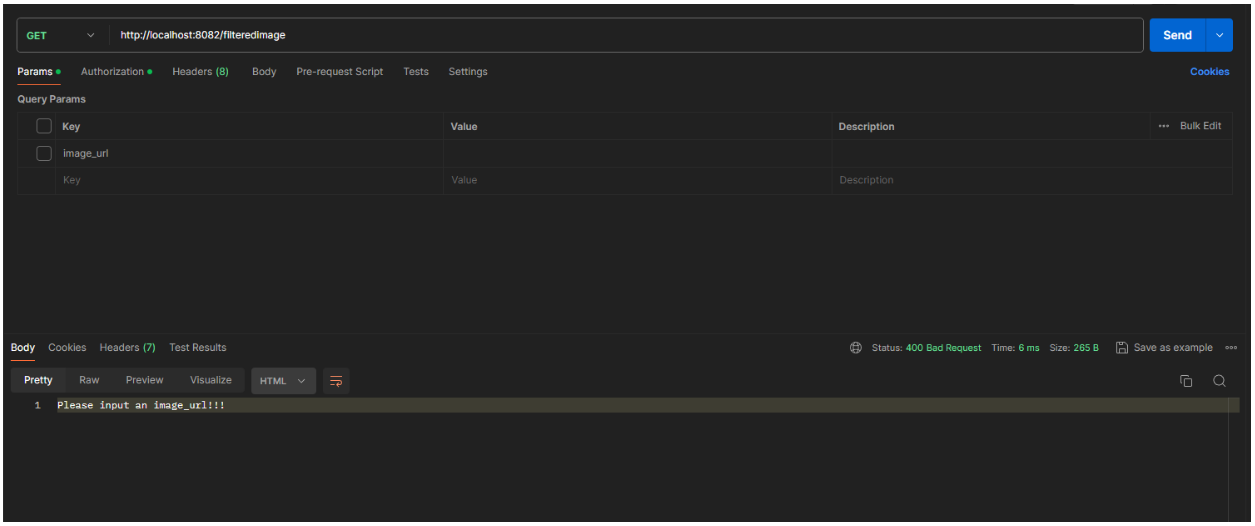


```
app.get("/filteredimage", async (req, res) => {
  const image_url = req.query.image_url;
  // 1. validate the image_url query. My api should only allow image urls start with http or https.
  if (image_url == null) {
    res.status(400).send("Please input an image_url!!!");
  }
  if (/^(http|https):\/\//i.test(image_url) === false) {
    res
      .status(400)
      .send("Please input a image_url that starts with http or https!!!");
  }
  //  2. call filterImageFromURL(image_url) to filter the image
  const imageLocal = await filterImageFromURL(image_url.toString());
  //  3. send the resulting file in the response and call a call back function to delete the file on the server
  res.status(200).sendFile(imageLocal, () => {
    //  4. deletes files image on this server.
    deleteLocalFiles([imageLocal]);
  });
});
```



```
ADMIN@LAPTOP-GG86CK8D MINGW64 ~/Documents/Image-Processing-Microservice-on-AWS
$ npm run dev

> api@1.0.0 dev
> node server.js

server running http://localhost:8082
press CTRL+C to stop server
```

Case 1: Image_url is null



http://localhost:8082/filteredimage

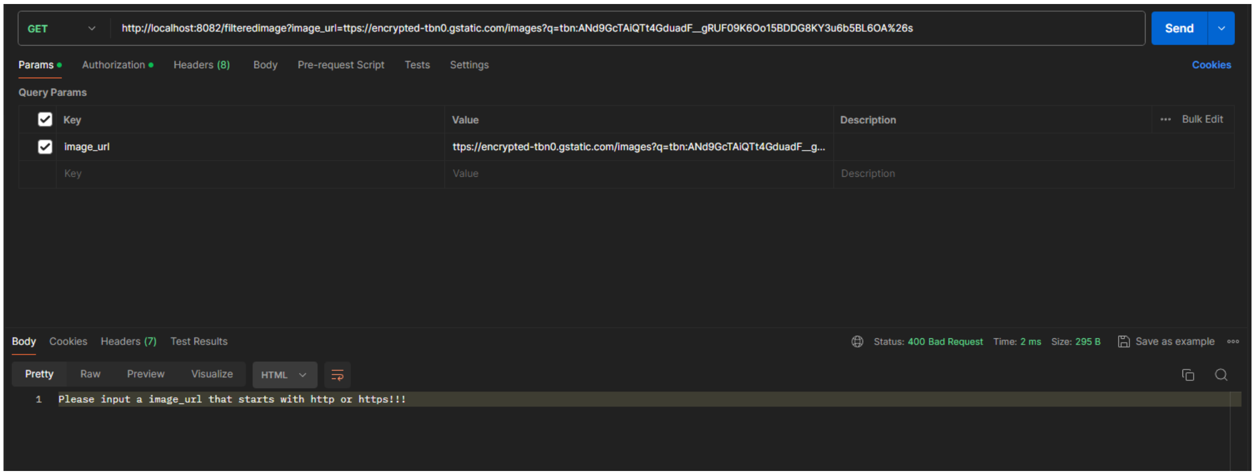
Query Params

	Key	Value	Description
□	image_url		
	Key	Value	Description

Status: 400 Bad Request Time: 6 ms Size: 265 B

```
Please input an image_url!!!
```

Case 2: Image_url is invalid (My api should only allow image urls start with http or https.)



http://localhost:8082/filteredimage?image_url=ttps://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTAiQTt4GduadF__gRUF09K6Oo15BDDG8KY3u6b5BL6OA%26s

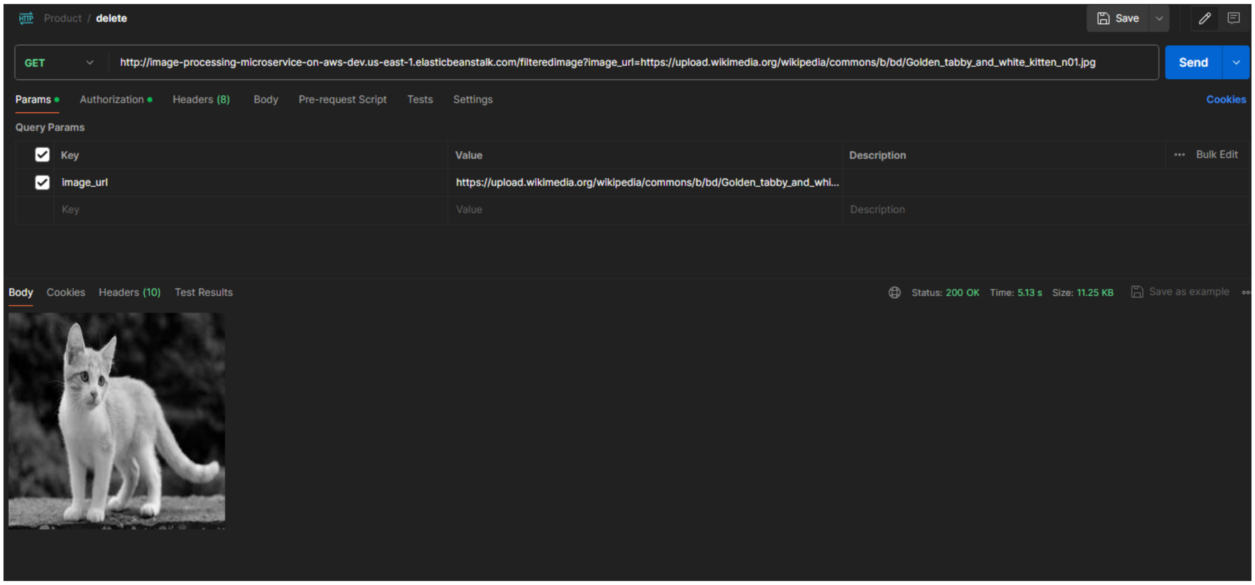
Query Params

☑	Key	Value	Description
☑	image_url	ttps://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTAiQTt4GduadF__g...	
	Key	Value	Description

Status: 400 Bad Request Time: 2 ms Size: 295 B

```
Please input a image_url that starts with http or https!!!
```

Case 3: Success query an image_url and receive status code 200:



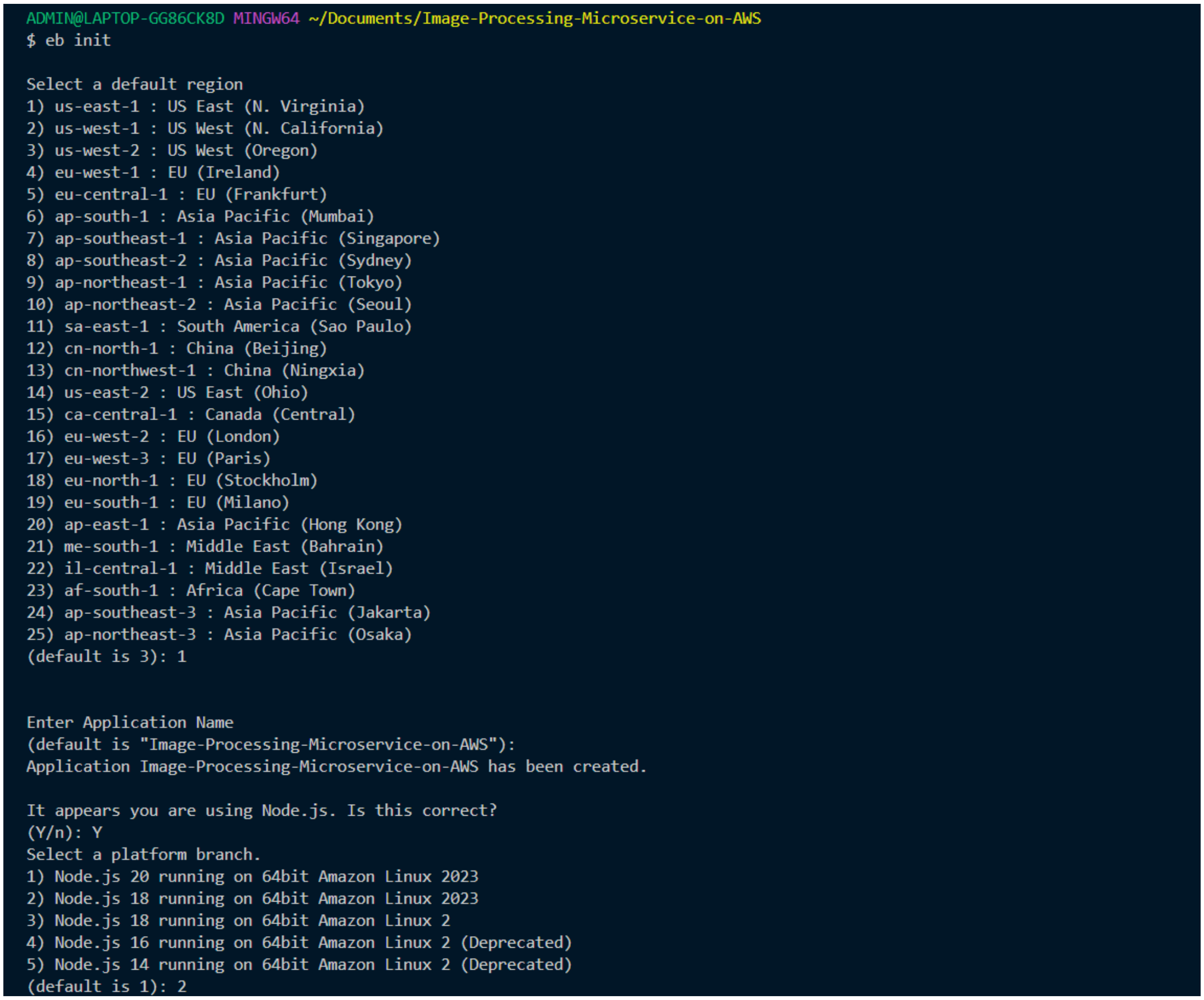
GET http://image-processing-microservice-on-aws-dev.us-east-1.elasticbeanstalk.com/filteredimage?image_url=https://upload.wikimedia.org/wikipedia/commons/b/bd/Golden_tabby_and_white_kitten_n01.jpg

Query Params

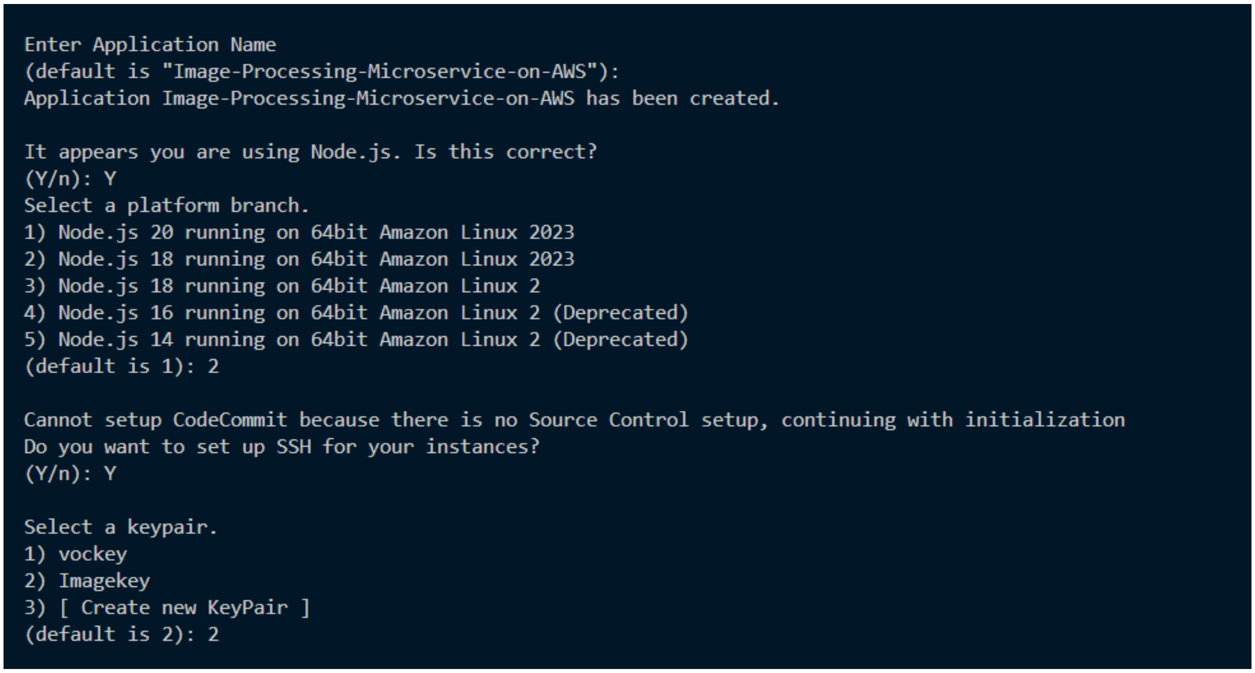
	Key	Value	Description
☑	image_url	https://upload.wikimedia.org/wikipedia/commons/b/bd/Golden_tabby_and_whi...	
	Key	Value	Description

Status: 200 OK Time: 5.13 s Size: 11.25 KB

- Step 3: Deploy my server to Elastic Beanstalk:
    - Eb init:

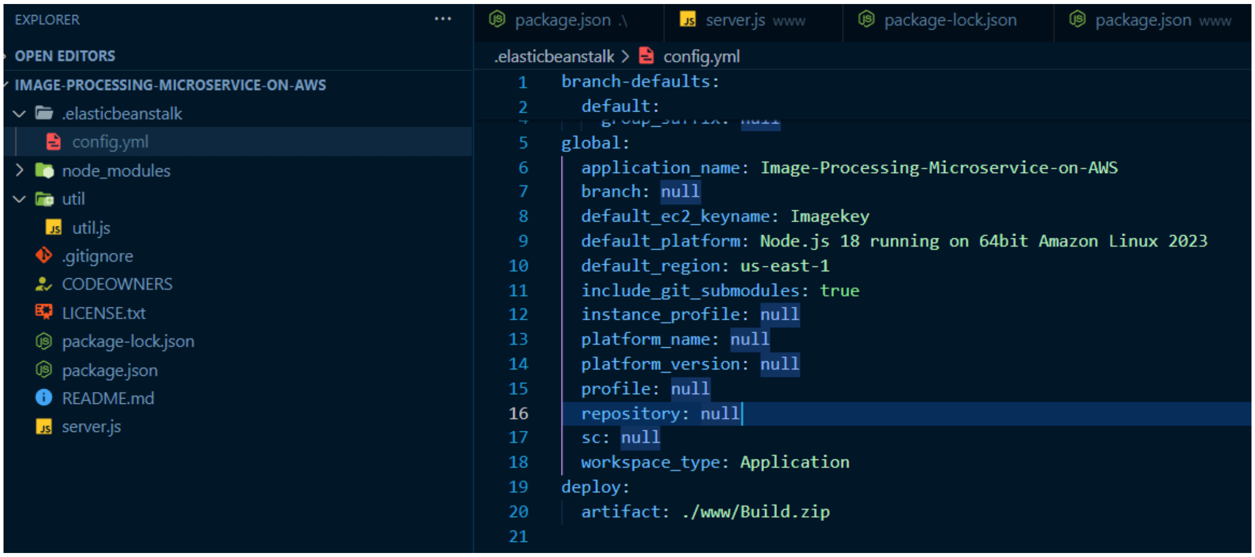


```
ADMIN@LAPTOP-GG86CK8D MINGW64 ~/Documents/Image-Processing-Microservice-on-AWS
$ eb init

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) il-central-1 : Middle East (Israel)
23) af-south-1 : Africa (Cape Town)
24) ap-southeast-3 : Asia Pacific (Jakarta)
25) ap-northeast-3 : Asia Pacific (Osaka)
(default is 3): 1


Enter Application Name
(default is "Image-Processing-Microservice-on-AWS"):
Application Image-Processing-Microservice-on-AWS has been created.

It appears you are using Node.js. Is this correct?
(Y/n): Y
Select a platform branch.
1) Node.js 20 running on 64bit Amazon Linux 2023
2) Node.js 18 running on 64bit Amazon Linux 2023
3) Node.js 18 running on 64bit Amazon Linux 2
4) Node.js 16 running on 64bit Amazon Linux 2 (Deprecated)
5) Node.js 14 running on 64bit Amazon Linux 2 (Deprecated)
(default is 1): 2
```



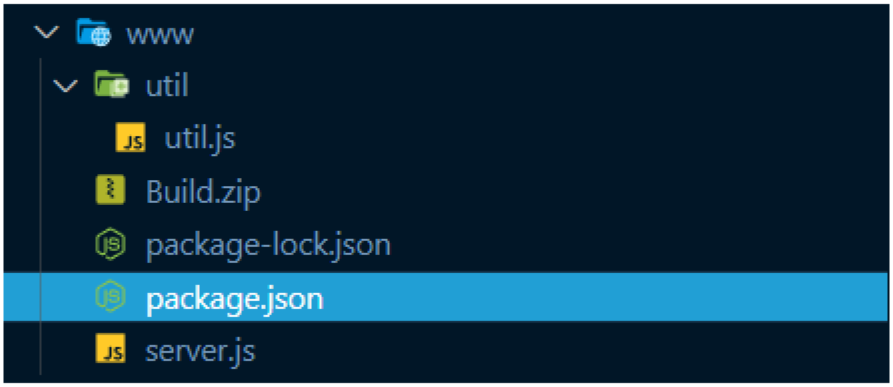
```
Enter Application Name
(default is "Image-Processing-Microservice-on-AWS"):
Application Image-Processing-Microservice-on-AWS has been created.

It appears you are using Node.js. Is this correct?
(Y/n): Y
Select a platform branch.
1) Node.js 20 running on 64bit Amazon Linux 2023
2) Node.js 18 running on 64bit Amazon Linux 2023
3) Node.js 18 running on 64bit Amazon Linux 2
4) Node.js 16 running on 64bit Amazon Linux 2 (Deprecated)
5) Node.js 14 running on 64bit Amazon Linux 2 (Deprecated)
(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) vockey
2) Imagekey
3) [ Create new KeyPair ]
(default is 2): 2
```

- Config deploy info: After eb init, a folder named .elasticbeanstalk is created and inside is a config.yml file. I will perform the deployment configuration with this file.

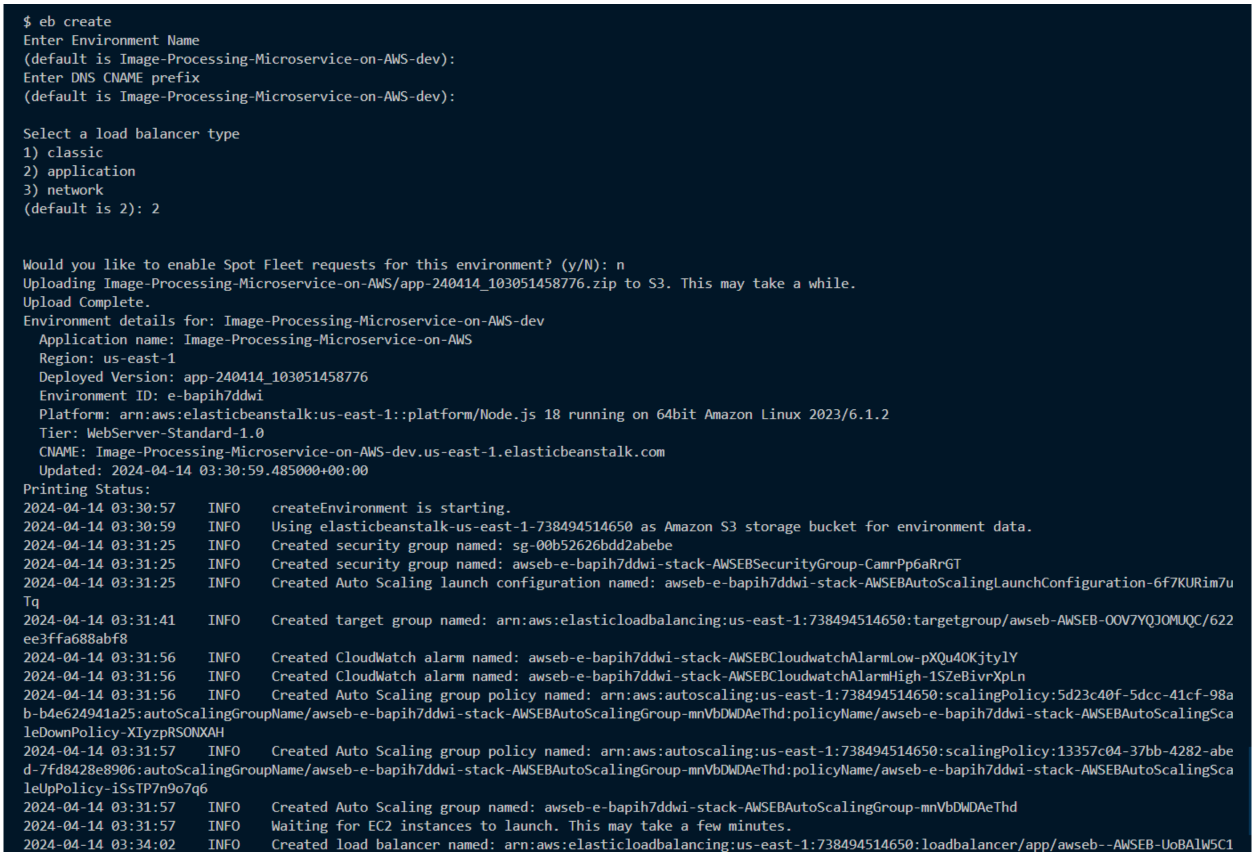


```
branch-defaults:
  default:
    group_suffix: null
global:
  application_name: Image-Processing-Microservice-on-AWS
  branch: null
  default_ec2_keyname: Imagekey
  default_platform: Node.js 18 running on 64bit Amazon Linux 2023
  default_region: us-east-1
  include_git_submodules: true
  instance_profile: null
  platform_name: null
  platform_version: null
  profile: null
  repository: null
  sc: null
  workspace_type: Application
deploy:
  artifact: ./www/Build.zip
```

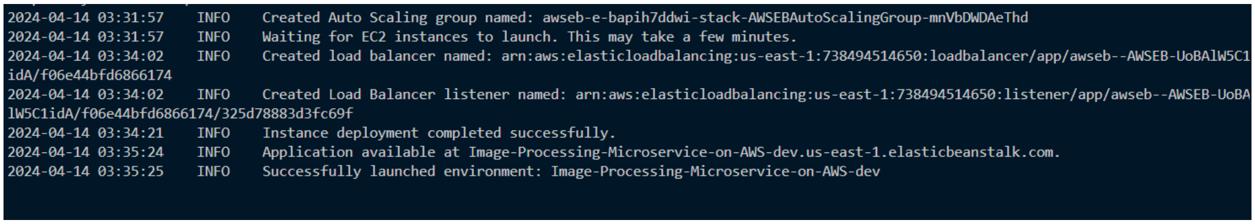
- Create folder www and file Build.zip (Includes server.js file, util folder, package.json, package-lock.json.)



- Eb create:



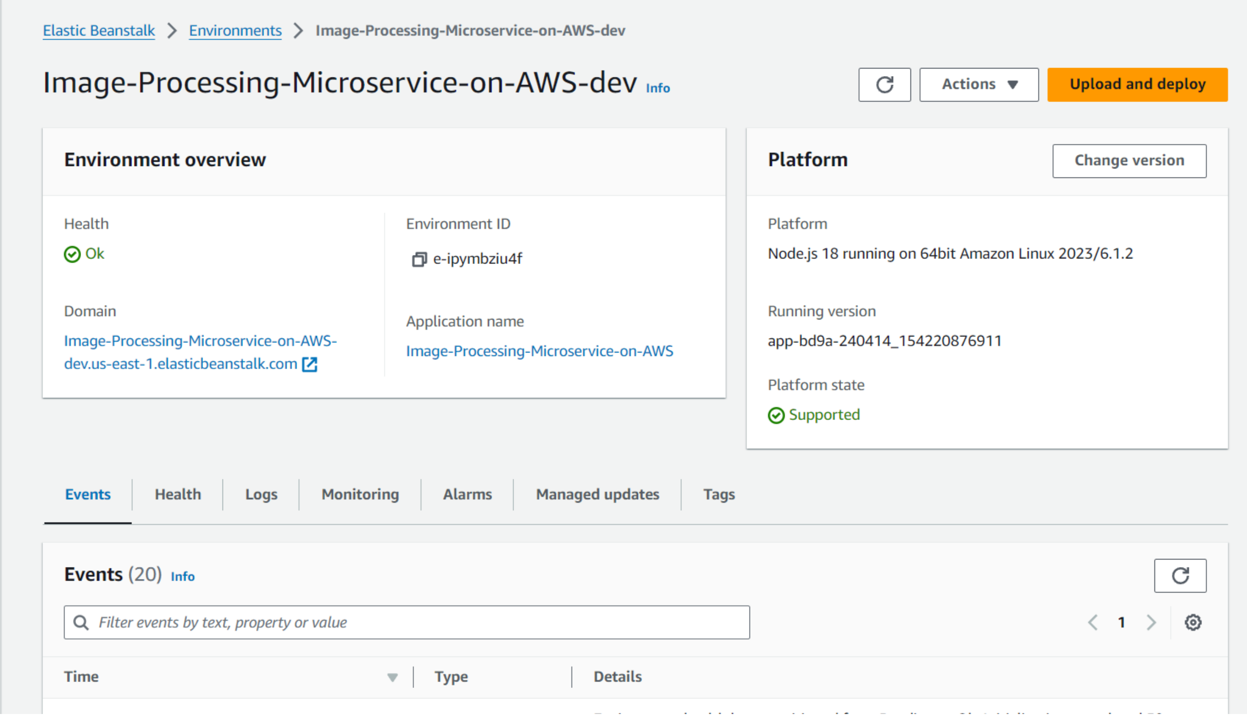
```
$ eb create
Enter Environment Name
(default is Image-Processing-Microservice-on-AWS-dev):
Enter DNS CNAME prefix
(default is Image-Processing-Microservice-on-AWS-dev):

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


Would you like to enable Spot Fleet requests for this environment? (y/N): n
Uploading Image-Processing-Microservice-on-AWS/app-240414_103051458776.zip to S3. This may take a while.
Upload Complete.
Environment details for: Image-Processing-Microservice-on-AWS-dev
  Application name: Image-Processing-Microservice-on-AWS
  Region: us-east-1
  Deployed Version: app-240414_103051458776
  Environment ID: e-bapih7ddwi
  Platform: arn:aws:elasticbeanstalk:us-east-1::platform/Node.js 18 running on 64bit Amazon Linux 2023/6.1.2
  Tier: WebServer-Standard-1.0
  CNAME: Image-Processing-Microservice-on-AWS-dev.us-east-1.elasticbeanstalk.com
  Updated: 2024-04-14 03:30:59.485000+00:00
Printing Status:
2024-04-14 03:30:57    INFO    createEnvironment is starting.
2024-04-14 03:30:59    INFO    Using elasticbeanstalk-us-east-1-738494514650 as Amazon S3 storage bucket for environment data.
2024-04-14 03:31:25    INFO    Created security group named: sg-00b52626bdd2abebe
2024-04-14 03:31:25    INFO    Created security group named: awseb-e-bapih7ddwi-stack-AWSEBSecurityGroup-CamrPp6aRrGT
2024-04-14 03:31:25    INFO    Created Auto Scaling launch configuration named: awseb-e-bapih7ddwi-stack-AWSEBAutoScalingLaunchConfiguration-6f7KURim7u
Tq
2024-04-14 03:31:41    INFO    Created target group named: arn:aws:elasticloadbalancing:us-east-1:738494514650:targetgroup/awseb-AWSEB-OOV7YQJOMUQC/622
ee3ffa688abf8
2024-04-14 03:31:56    INFO    Created CloudWatch alarm named: awseb-e-bapih7ddwi-stack-AWSEBCloudwatchAlarmLow-pXQu4OKjtylY
2024-04-14 03:31:56    INFO    Created CloudWatch alarm named: awseb-e-bapih7ddwi-stack-AWSEBCloudwatchAlarmHigh-1SZeBivrXpLn
2024-04-14 03:31:56    INFO    Created Auto Scaling group policy named: arn:aws:autoscaling:us-east-1:738494514650:scalingPolicy:5d23c40f-5dcc-41cf-98a
b-b4e624941a25:autoScalingGroupName/awseb-e-bapih7ddwi-stack-AWSEBAutoScalingGroup-mnVbDWDAeThd:policyName/awseb-e-bapih7ddwi-stack-AWSEBAutoScalingSca
leDownPolicy-XIyzpRSONXAH
2024-04-14 03:31:57    INFO    Created Auto Scaling group policy named: arn:aws:autoscaling:us-east-1:738494514650:scalingPolicy:13357c04-37bb-4282-abe
d-7fd8428e8906:autoScalingGroupName/awseb-e-bapih7ddwi-stack-AWSEBAutoScalingGroup-mnVbDWDAeThd:policyName/awseb-e-bapih7ddwi-stack-AWSEBAutoScalingSca
leUpPolicy-iSsTP7n9o7q6
2024-04-14 03:31:57    INFO    Created Auto Scaling group named: awseb-e-bapih7ddwi-stack-AWSEBAutoScalingGroup-mnVbDWDAeThd
2024-04-14 03:31:57    INFO    Waiting for EC2 instances to launch. This may take a few minutes.
2024-04-14 03:34:02    INFO    Created load balancer named: arn:aws:elasticloadbalancing:us-east-1:738494514650:loadbalancer/app/awseb--AWSEB-UoBAlW5C1
```



```
2024-04-14 03:31:57    INFO    Created Auto Scaling group named: awseb-e-bapih7ddwi-stack-AWSEBAutoScalingGroup-mnVbDWDAeThd
2024-04-14 03:31:57    INFO    Waiting for EC2 instances to launch. This may take a few minutes.
```

```
2024-04-14 03:34:02    INFO    Created load balancer named: arn:aws:elasticloadbalancing:us-east-1:738494514650:loadbalancer/app/awseb--AWSEB-UoBAlW5C1
idA/f06e44bfd6866174
2024-04-14 03:34:02    INFO    Created Load Balancer listener named: arn:aws:elasticloadbalancing:us-east-1:738494514650:listener/app/awseb--AWSEB-UoBA
lW5C1idA/f06e44bfd6866174/325d78883d3fc69f
2024-04-14 03:34:21    INFO    Instance deployment completed successfully.
2024-04-14 03:35:24    INFO    Application available at Image-Processing-Microservice-on-AWS-dev.us-east-1.elasticbeanstalk.com.
2024-04-14 03:35:25    INFO    Successfully launched environment: Image-Processing-Microservice-on-AWS-dev
```

- Check result deploy:



Elastic Beanstalk > Environments > Image-Processing-Microservice-on-AWS-dev

Image-Processing-Microservice-on-AWS-dev Info

Environment overview

Health
Ok

Environment ID
e-ipymbziu4f

Domain
Image-Processing-Microservice-on-AWS-dev.us-east-1.elasticbeanstalk.com

Application name
Image-Processing-Microservice-on-AWS

Platform

Platform
Node.js 18 running on 64bit Amazon Linux 2023/6.1.2

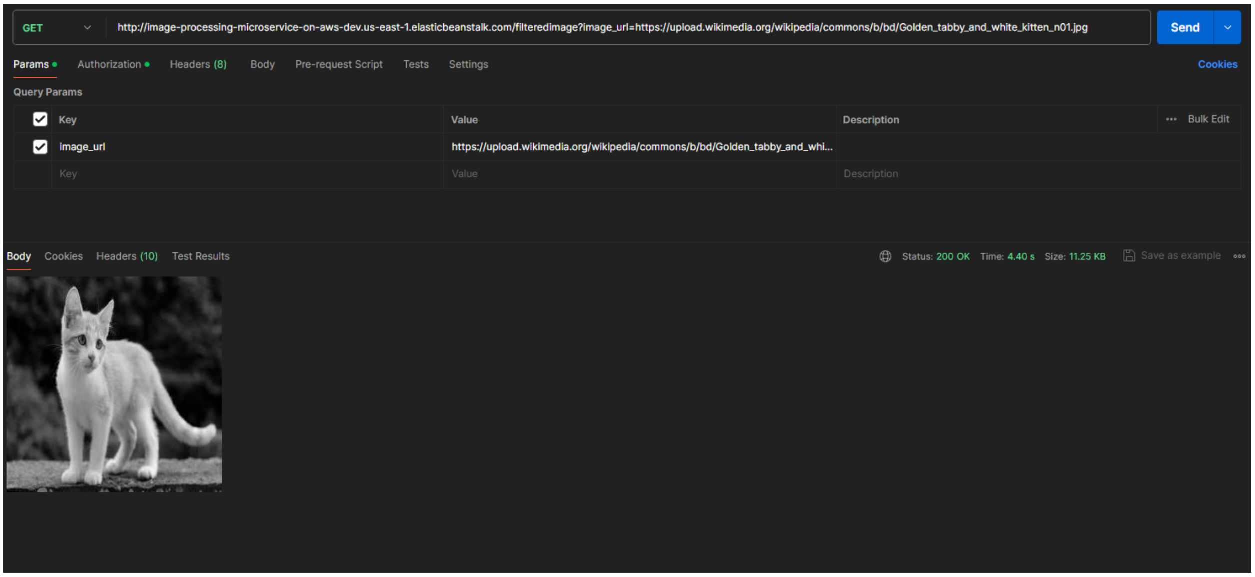
Running version
app-bd9a-240414_154220876911

Platform state
Supported

Events (20) Info

Time	Type	Details

- Test api with postman:

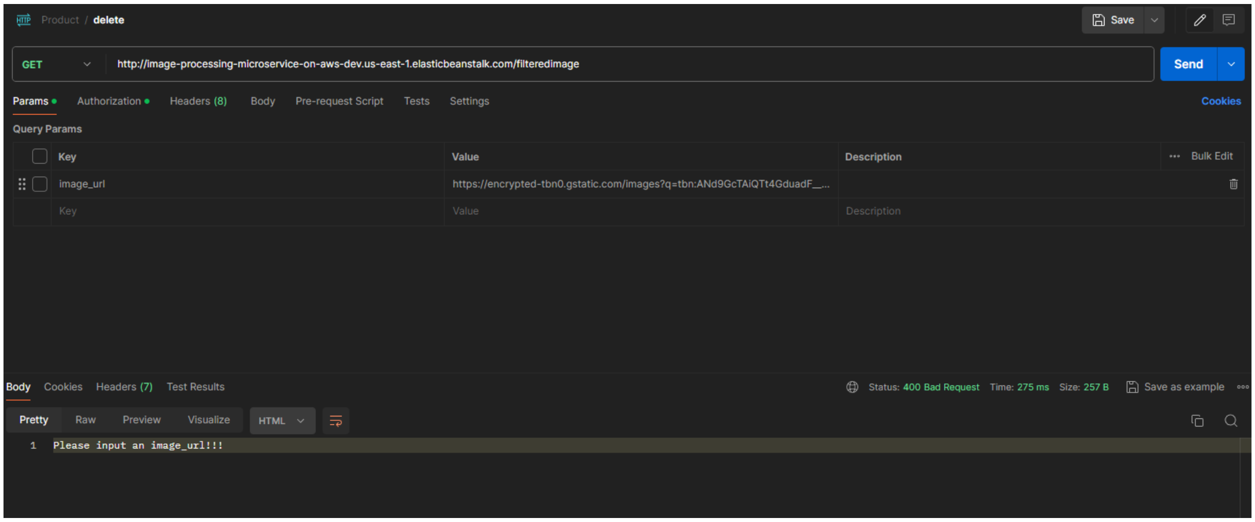


GET http://image-processing-microservice-on-aws-dev.us-east-1.elasticbeanstalk.com/filteredimage?image_url=https://upload.wikimedia.org/wikipedia/commons/b/bd/Golden_tabby_and_white_kitten_n01.jpg

Query Params

☑	Key	Value	Description
☑	image_url	https://upload.wikimedia.org/wikipedia/commons/b/bd/Golden_tabby_and_whi...	
	Key	Value	Description

Status: 200 OK Time: 4.40 s Size: 11.25 KB



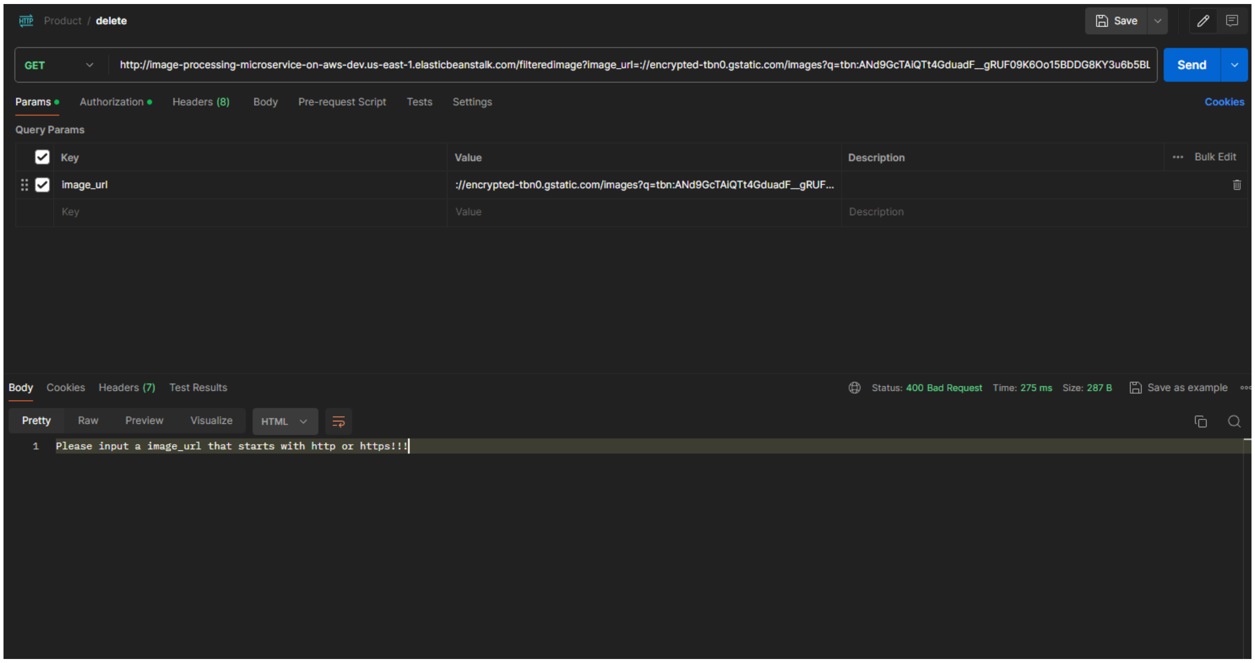
GET http://image-processing-microservice-on-aws-dev.us-east-1.elasticbeanstalk.com/filteredimage

Query Params

	Key	Value	Description
□	image_url	https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTAiQTt4GduadF__...	
	Key	Value	Description

Status: 400 Bad Request Time: 275 ms Size: 257 B

```
Please input an image_url!!!
```



GET http://image-processing-microservice-on-aws-dev.us-east-1.elasticbeanstalk.com/filteredimage?image_url=://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTAiQTt4GduadF__gRUF09K6Oo15BDDG8KY3u6b5BL

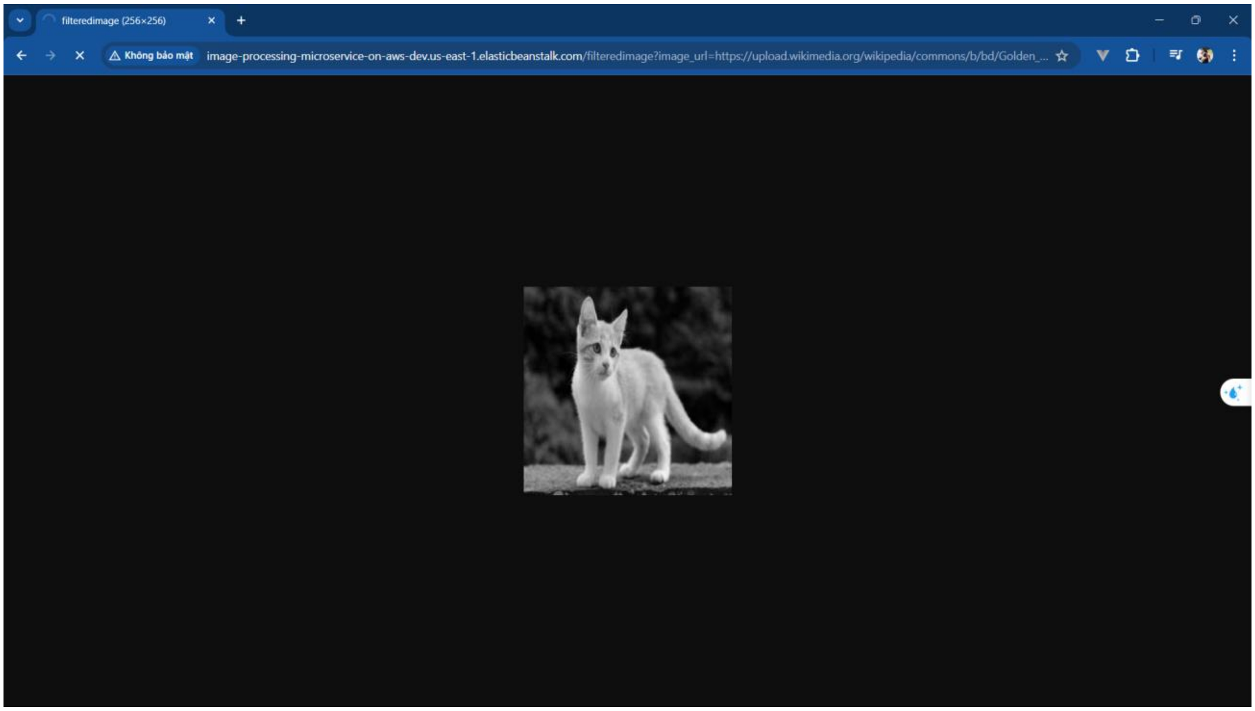
Query Params

	Key	Value	Description
☑	image_url	://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTAiQTt4GduadF__gRUF...	
	Key	Value	Description

Status: 400 Bad Request Time: 275 ms Size: 287 B

```
Please input a image_url that starts with http or https!!!
```

- Test website:


filteredimage (256×256)
Không bảo mật
image-processing-microservice-on-aws-dev.us-east-1.elasticbeanstalk.com/filteredimage?image_url=https://upload.wikimedia.org/wikipedia/commons/b/bd/Golden_...

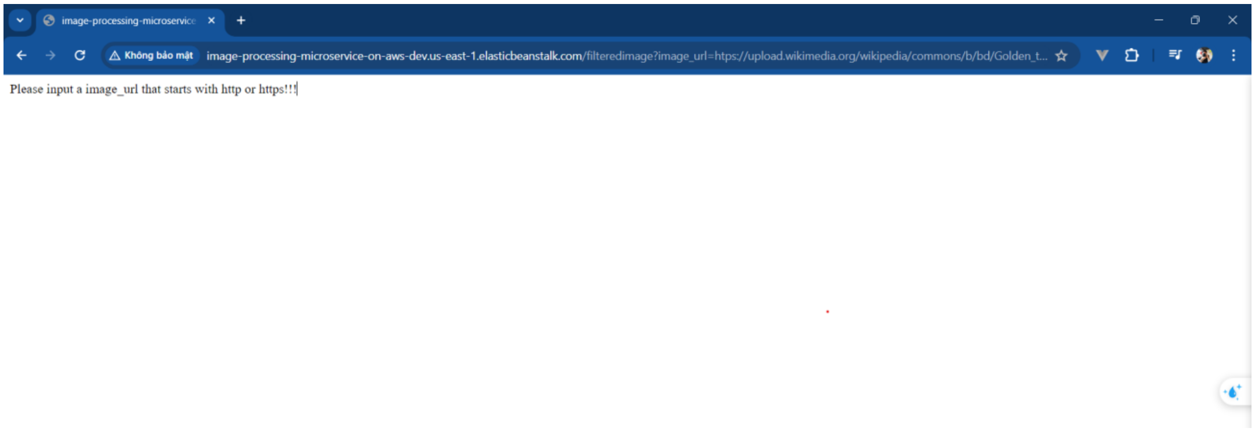


image-processing-microservice
Không bảo mật
image-processing-microservice-on-aws-dev.us-east-1.elasticbeanstalk.com/filteredimage?image_url=https://upload.wikimedia.org/wikipedia/commons/b/bd/Golden_t...

Please input a image_url that starts with http or https!!!

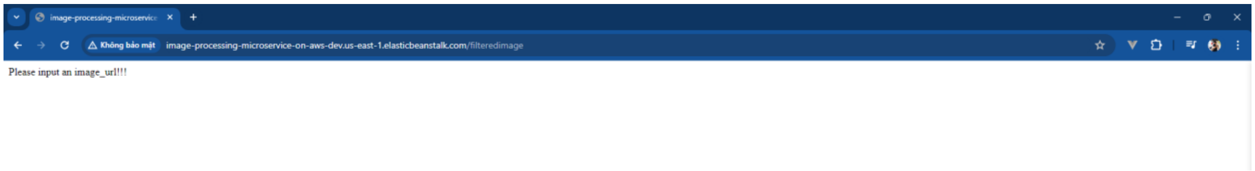


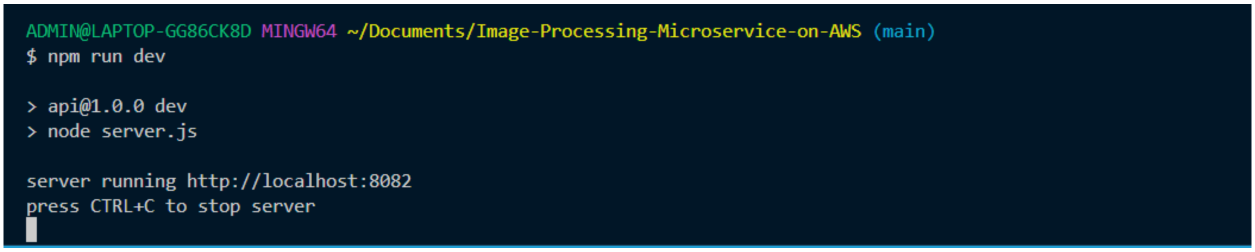
image-processing-microservice
Không bảo mật
image-processing-microservice-on-aws-dev.us-east-1.elasticbeanstalk.com/filteredimage

Please input an image_url!!!

Finally: This is my link to the endpoint URL for a running elastic beanstalk deployment: Image processing deployment url

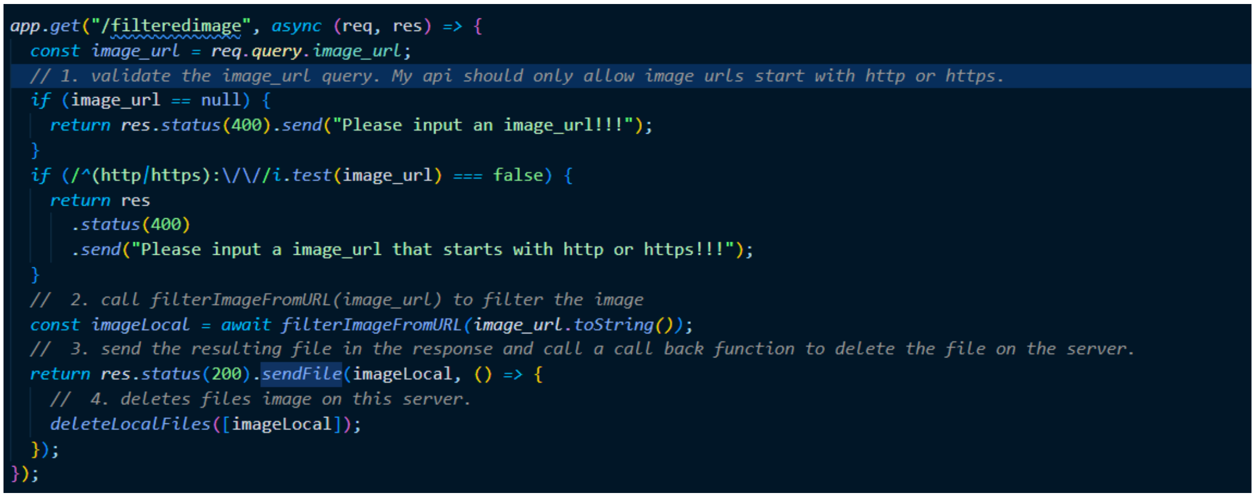
Project Rubric:

- Starting the server with `npm run dev` runs a local instance of the server with no errors:

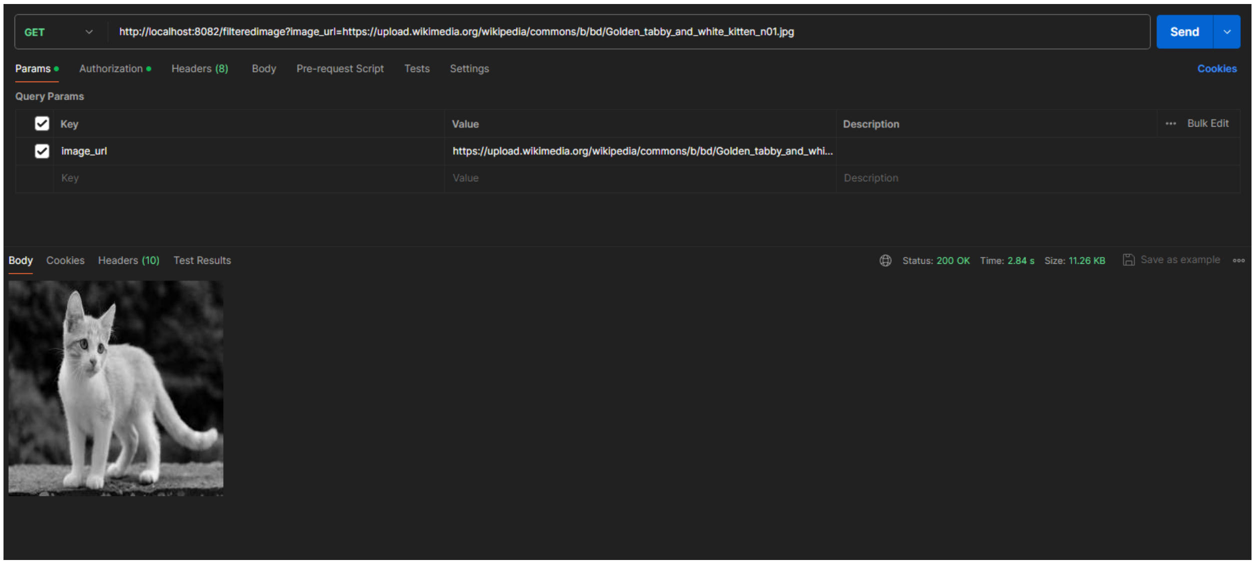


```
ADMIN@LAPTOP-GG86CK8D MINGW64 ~/Documents/Image-Processing-Microservice-on-AWS (main)
$ npm run dev

> api@1.0.0 dev
> node server.js

server running http://localhost:8082
press CTRL+C to stop server
```

- The stubbed `@TODO1` endpoint in `src/server.js` is completed and accepts valid requests:



```
app.get("/filteredimage", async (req, res) => {
  const image_url = req.query.image_url;
  // 1. validate the image_url query. My api should only allow image urls start with http or https.
  if (image_url == null) {
    return res.status(400).send("Please input an image_url!!!");
  }
  if (/^(http|https):\/\//i.test(image_url) === false) {
    return res
      .status(400)
      .send("Please input a image_url that starts with http or https!!!");
  }
  //  2. call filterImageFromURL(image_url) to filter the image
  const imageLocal = await filterImageFromURL(image_url.toString());
  //  3. send the resulting file in the response and call a call back function to delete the file on the server.
  return res.status(200).sendFile(imageLocal, () => {
    //  4. deletes files image on this server.
    deleteLocalFiles([imageLocal]);
  });
});
```

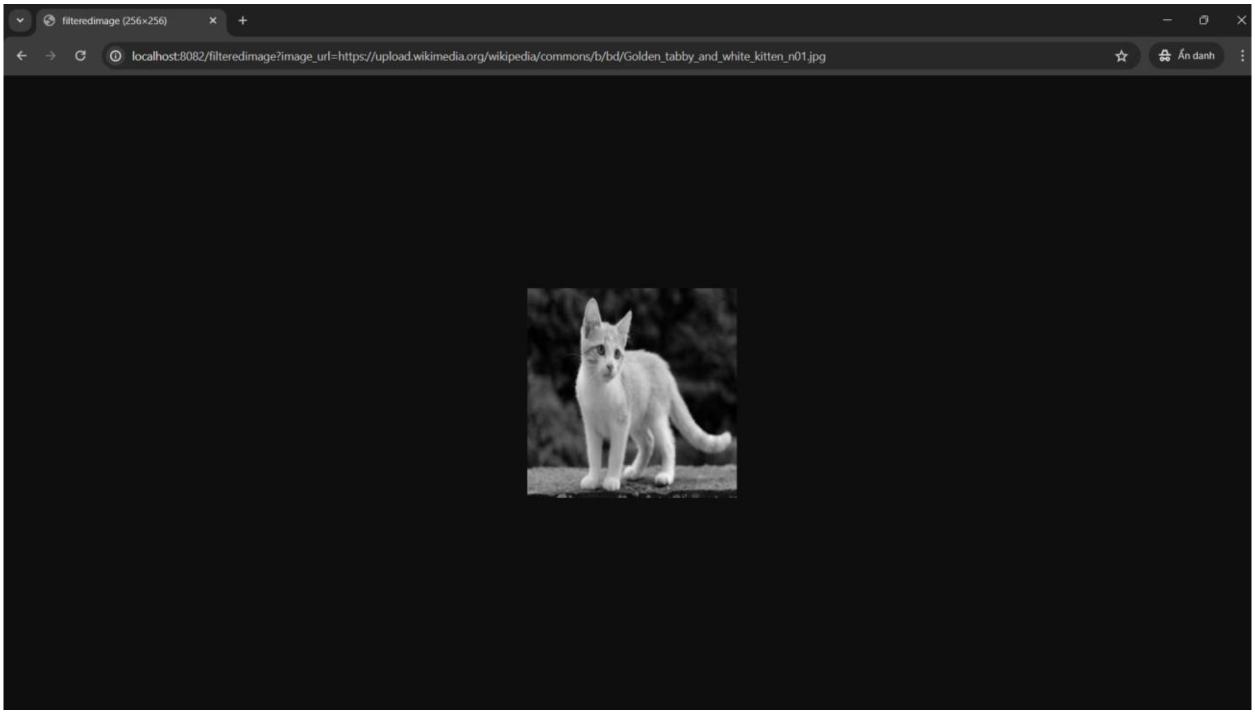


GET http://localhost:8082/filteredimage?image_url=https://upload.wikimedia.org/wikipedia/commons/b/bd/Golden_tabby_and_white_kitten_n01.jpg

Query Params

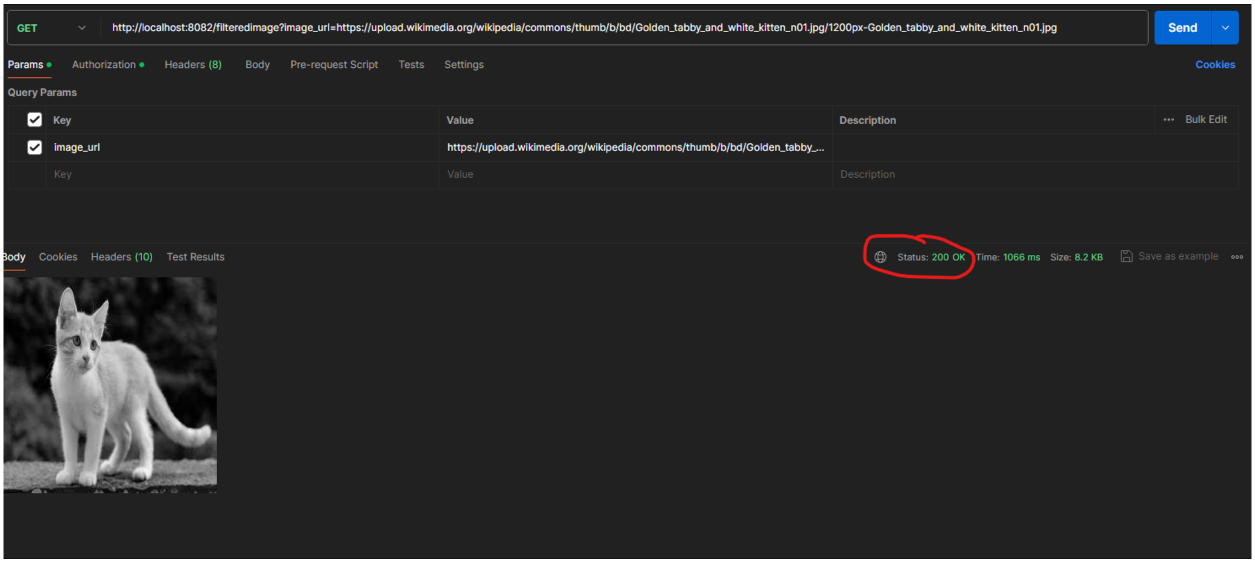
☑	Key	Value	Description
☑	image_url	https://upload.wikimedia.org/wikipedia/commons/b/bd/Golden_tabby_and_whi...	
	Key	Value	Description

Status: 200 OK Time: 2.84 s Size: 11.26 KB



localhost:8082/filteredimage?image_url=https://upload.wikimedia.org/wikipedia/commons/b/bd/Golden_tabby_and_white_kitten_n01.jpg

- Successful responses have a 200 code, at least one error code for caught errors (i.e. 422):

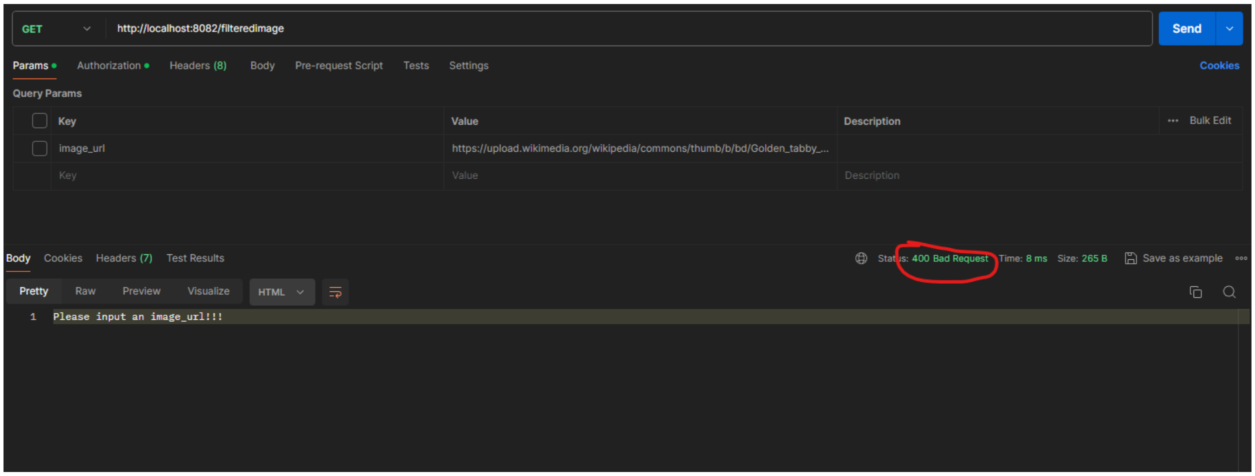


GET http://localhost:8082/filteredimage?image_url=https://upload.wikimedia.org/wikipedia/commons/thumb/b/bd/Golden_tabby_and_white_kitten_n01.jpg/1200px-Golden_tabby_and_white_kitten_n01.jpg

Query Params

	Key	Value	Description
☑	image_url	https://upload.wikimedia.org/wikipedia/commons/thumb/b/bd/Golden_tabby_...	
	Key	Value	Description

Status: 200 OK Time: 1066 ms Size: 8.2 KB



http://localhost:8082/filteredimage

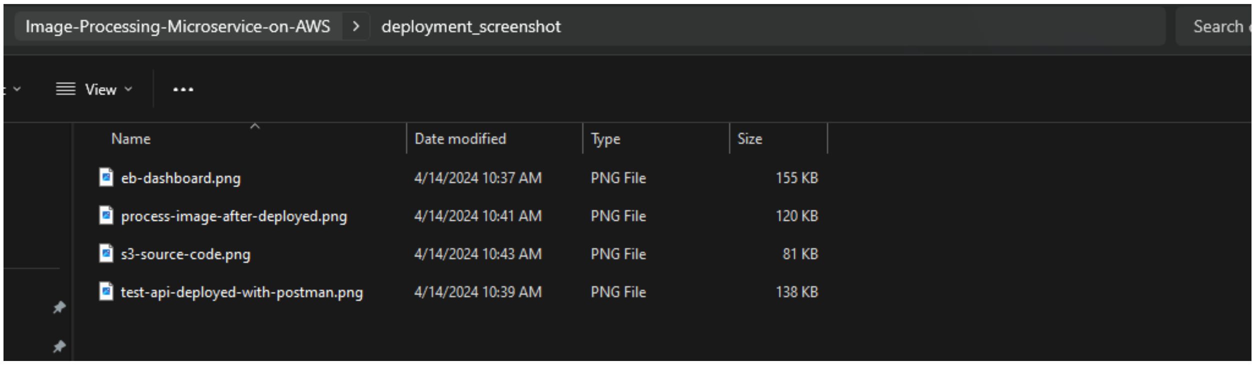
Query Params

	Key	Value	Description
□	image_url	https://upload.wikimedia.org/wikipedia/commons/thumb/b/bd/Golden_tabby_...	
	Key	Value	Description

Status: 400 Bad Request Time: 8 ms Size: 265 B

```
Please input an image_url!!!
```

- The project was deployed using the AWS Elastic Beanstalk CLI `eb init`, `eb create`, and `eb deploy` commands: I have list at Step 3.
- A screenshot of the elastic beanstalk application dashboard is included in a `deployment_screenshot` director:



Name	Date modified	Type	Size
eb-dashboard.png	4/14/2024 10:37 AM	PNG File	155 KB
process-image-after-deployed.png	4/14/2024 10:41 AM	PNG File	120 KB
s3-source-code.png	4/14/2024 10:43 AM	PNG File	81 KB
test-api-deployed-with-postman.png	4/14/2024 10:39 AM	PNG File	138 KB

- An endpoint URL for a running elastic beanstalk deployment (EB_URL) has been submitted along with the project submission. This endpoint responds to valid GET requests: http://image-processing-microservice-on-aws-dev.us-east-1.elasticbeanstalk.com/filteredimage?image_url=https://upload.wikimedia.org/wikipedia/commons/b/bd/Golden_tabby_and_white_kitten_n01.jpg

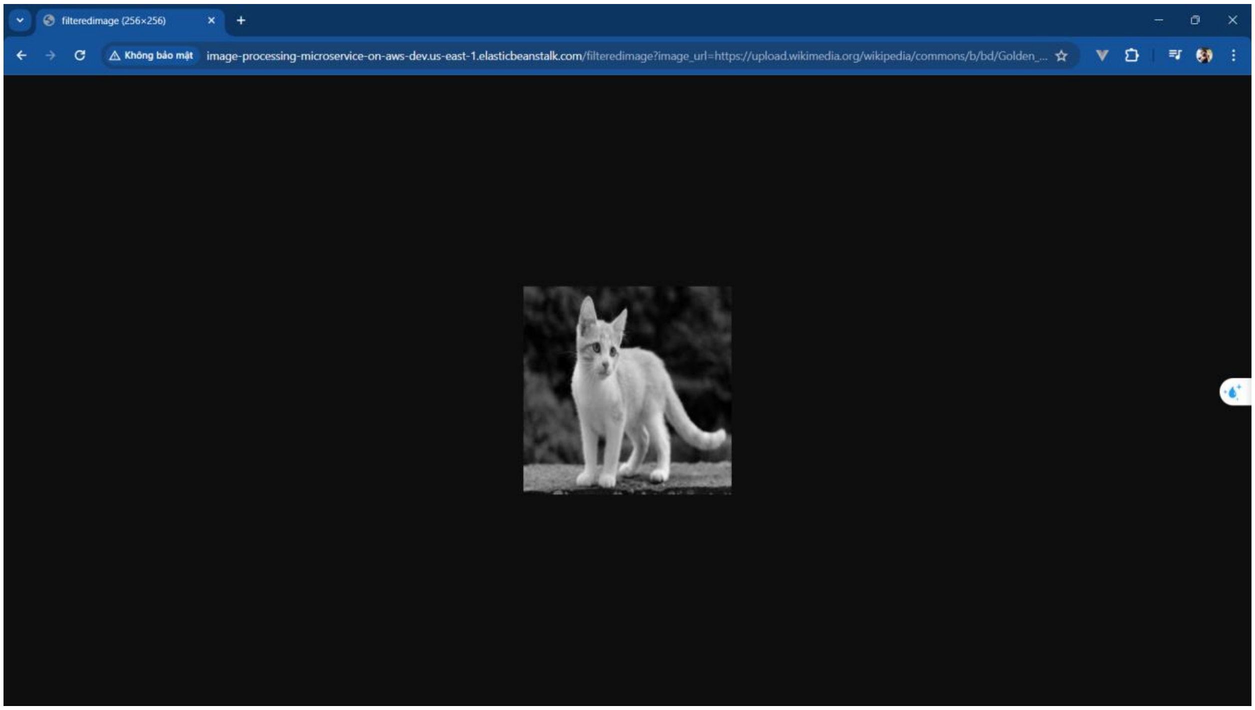


image-processing-microservice-on-aws-dev.us-east-1.elasticbeanstalk.com/filteredimage?image_url=https://upload.wikimedia.org/wikipedia/commons/b/bd/Golden_...

THANK ALL OF YOU FOR READING MY REPORT!!!