

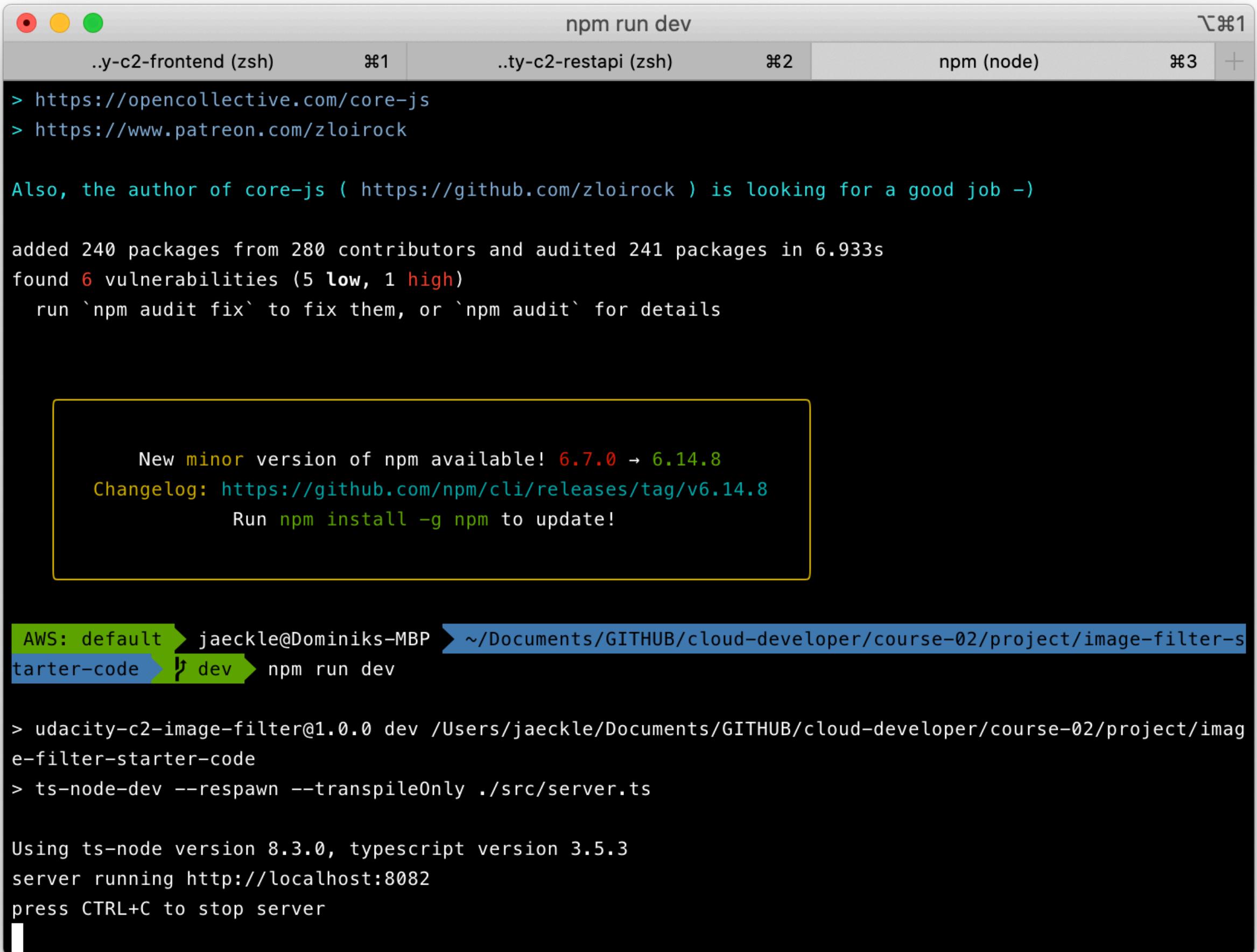
Udacity Project 02

Udagram: Your Own Instagram on AWS

GITHUB REPOSITORY: <https://github.com/dominikjaeckle/cloud-developer/tree/master>

Dominik Jäckle | August 23, 2020

Server runs without errors



A screenshot of a macOS terminal window titled "npm run dev". The window has three tabs: ".y-c2-frontend (zsh)", ".ty-c2-restapi (zsh)", and "npm (node)". The "npm (node)" tab is active, showing the following output:

```
> https://opencollective.com/core-js
> https://www.patreon.com/zloirock

Also, the author of core-js ( https://github.com/zloirock ) is looking for a good job -)

added 240 packages from 280 contributors and audited 241 packages in 6.933s
found 6 vulnerabilities (5 low, 1 high)
  run `npm audit fix` to fix them, or `npm audit` for details
```

New minor version of npm available! 6.7.0 → 6.14.8
Changelog: <https://github.com/npm/cli/releases/tag/v6.14.8>
Run npm install -g npm to update!

The terminal prompt shows "AWS: default > jaeckle@Dominiks-MBP > ~/Documents/GITHUB/cloud-developer/course-02/project/image-filter-starter-code > dev > npm run dev". The command "npm run dev" was run, and the output shows the server is running at `http://localhost:8082`.

Filter endpoint works for localhost

The screenshot shows the Postman application interface. In the left sidebar, under 'Collections', the 'udacity-project2 udagram' collection is selected. Inside it, a request titled 'GET {{host}}/filteredimage' is highlighted. The request URL is set to '{{host}}/filteredimage?image_url=https://upload.wikimedia.org/wikipedia/commons/b/bd/Golden_tabby_and_white_kitten.jpg'. The 'Params' tab shows a single parameter 'image_url' with the value 'https://upload.wikimedia.org/wikipedia/commons/b/bd/Golden_tabby_and_white_kitten.jpg'. The 'Body' tab displays a black and white image of a kitten. At the bottom of the interface, the status bar indicates 'Status: 200 OK'.

Filter endpoint cannot load image due to typo in URL.

The screenshot shows the Postman application interface. On the left, the sidebar displays 'Collections' (udacity-c2-basic-server, udacity-c2-restapi, udacity-project2 udagram) and a selected 'GET {{host}}/filteredimage'. The main workspace shows a 'GET {{host}}/filteredimage' request with the URL {{host}}/filteredimage?image_url=https://upload.wikimedia.org/wikipedia/commons/b/bd/Golden_tabby_and_white_...'. The 'Params' tab is active, showing a single parameter 'image_url' with the value https://upload.wikimedia.org/wikipedia/commons/b/bd/Golden_tabby_and_white.... The response section indicates a 'Status: 500 Internal Server Error' with the message 'Cannot process image file. Please double check the file URL.'

Robustness

```
app.get("/filteredimage/", ( req: Request, res: Response ) => {
  let { image_url } = req.query;
  if (!image_url) {
    return res.status(400).send(`Please provide a url as parameter`);
  }

  // validate the image url using a REGEX parser + a filetype checker
  const valid = isUrl(image_url) ? isImageFile(image_url) : false;
  if (!valid) {
    return res.status(400).send(`The provided URL ${image_url} is invalid.`);
  }

  // filter the image
  let result = filterImageFromURL(image_url).then((f: string) => {
    return res.status(200).sendFile(f, () => {
      deleteLocalFiles([f]);
    });
  }).catch((r: string) => {
    return res.status(500)
      .send("Cannot process image file. Please double check the file URL.");
  })
});
```

Check for parameter

URL validation + filetype validation

Extension of promise to check if image could be loaded

Extended util.ts

```
40  /**
41   * Helper function to validate an URL.
42   * Code taken from https://gist.github.com/rodneyrehm/8013067
43   * and then adapted to check the following protocols:
44   * - check protocol: http://, https://, file://, ftp://
45   *
46   * @param textUrl
47   */
48  export function isUrl(textUrl: string): boolean {
49    let url_pattern = /^(https?|ftp|file):\/\/(-\.)?([^\s\/?\.#]+\.\.?)?([^\s]*)?$/i;
50    if(textUrl.match(url_pattern)) {
51      return true;
52    } else {
53      return false;
54    }
55  }
56
57 /**
58  * Method checks for a respective filetype. This is done by only
59  * looking at the file type.
60  *
61  * @param textUrl
62  * @param filetypes
63  */
64  export function isImageFile(textUrl: string,
65    filetypes: string[] = ['jpg', 'jpeg', 'png']): boolean {
66    let urlSplit = textUrl.split('.');
67    let fileType = urlSplit[urlSplit.length - 1].trim();
68    return filetypes.includes(fileType);
69 }
```

eb init

```
jaeckle@Dominiks-MBP: ~/Documents/GITHUB/cloud-developer/course-02/project/image-filter-starter... ~%1
..y-c2-frontend (zsh)   %1 ..ty-c2-restapi (zsh)   %2 ..-starter-code (zsh)   %3 +
x AWS: default > jaeckle@Dominiks-MBP > ~/Documents/GITHUB/cloud-developer/course-02/project/image-filter-starter-code > dev 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): 4

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

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

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-ec2
2) udagram-dj-vid
3) [ Create new KeyPair ]
(default is 2): 1
```

```
TS server.ts ! config.yml image-filter-starter-code • .elasticbeanstalk X {} package.json

image-filter-starter-code > .elasticbeanstalk > ! config.yml
1 branch-defaults:
2   default:
3     | environment: null
4   deploy:
5     | artifact: ./www/Archive.zip
6   global:
7     | application_name: image-filter-starter-code
8     | branch: null
9     | default_ec2_keyname: udacity-ec2
10    | default_platform: Node.js running on 64bit Amazon Linux
11    | default_region: eu-west-1
12    | include_git_submodules: true
13    | instance_profile: null
14    | platform_name: null
15    | platform_version: null
16    | profile: null
17    | repository: null
18    | sc: null
19    | workspace_type: Application
20
```

npm run build

```
x ➤ AWS: default ➤ jaeckle@Dominiks-MBP ➤ ~/Documents/GITHUB/cloud-developer/course-02/project/image-filter-starter-code ➤ ↵ dev ➤ npm run build

> udacity-c2-image-filter@1.0.0 build /Users/jaeckle/Documents/GITHUB/cloud-developer/course-02/project/image-filter-starter-code
> npm run clean && tsc && cp package.json www/package.json && mkdir www/tmp/ && cd www && zip -r Archive.zip . && cd ..

> udacity-c2-image-filter@1.0.0 clean /Users/jaeckle/Documents/GITHUB/cloud-developer/course-02/project/image-filter-starter-code
> rm -rf www/ || true

    adding: util/ (stored 0%)
    adding: util/util.js (deflated 60%)
    adding: util/util.js.map (deflated 61%)
    adding: server.js (deflated 60%)
    adding: server.js.map (deflated 65%)
    adding: package.json (deflated 53%)
    adding: tmp/ (stored 0%)
AWS: default ➤ jaeckle@Dominiks-MBP ➤ ~/Documents/GITHUB/cloud-developer/course-02/project/image-filter-starter-code ➤ ↵ dev ➤
```

eb create → <http://image-filter-starter-code-dev2.eu-west-1.elasticbeanstalk.com>

```
eb create
Enter Environment Name
(default is image-filter-starter-code-dev):
Enter DNS CNAME prefix
(default is image-filter-starter-code-dev2):

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): n
Uploading image-filter-starter-code/app-200822_235853.zip to S3. This may take a while.
Upload Complete.

Environment details for: image-filter-starter-code-dev
  Application name: image-filter-starter-code
  Region: eu-west-1
  Deployed Version: app-200822_235853
  Environment ID: e-9dppjbdgin
  Platform: arn:aws:elasticbeanstalk:eu-west-1::platform/Node.js running on 64bit Amazon Linux/4.15.1
  Tier: WebServer-Standard-1.0
  CNAME: image-filter-starter-code-dev2.eu-west-1.elasticbeanstalk.com
  Updated: 2020-08-22 21:58:59.178000+00:00

Printing Status:
2020-08-22 21:58:57  INFO  createEnvironment is starting.
2020-08-22 21:58:59  INFO  Using elasticbeanstalk-eu-west-1-386524155188 as Amazon S3 storage bucket for environment data.
2020-08-22 21:59:19  INFO  Created target group named: arn:aws:elasticloadbalancing:eu-west-1:386524155188:targetgroup/awseb-AWSEB-KU2L6J2PF026/806afc37bd901955
2020-08-22 21:59:19  INFO  Created security group named: sg-0284317ca792055a3
2020-08-22 21:59:34  INFO  Created security group named: awseb-e-9dppjbdgin-stack-AWSEBSecurityGroup-1T00MV8E6WKTY
2020-08-22 21:59:34  INFO  Created Auto Scaling launch configuration named: awseb-e-9dppjbdgin-stack-AWSEBAutoScalingLaunchConfiguration-1RMX4IWOTAVQ
2020-08-22 22:00:51  INFO  Created Auto Scaling group named: awseb-e-9dppjbdgin-stack-AWSEBAutoScalingGroup-1ER22C20JYcba
2020-08-22 22:00:51  INFO  Waiting for EC2 instances to launch. This may take a few minutes.
2020-08-22 22:01:06  INFO  Created Auto Scaling group policy named: arn:aws:autoscaling:eu-west-1:386524155188:scalingPolicy:3f2520af-8ade-41fe-bbe9-134180a127e:autoScalingGroupName/awseb-e-9dppjbdgin-stack-AWSEBAutoScalingGroup-1ER22C20JYcba:policyName/awseb-e-9dppjbdgin-stack-AWSEBAutoScalingScaleUpPolicy-U9I6H69ED004
2020-08-22 22:01:06  INFO  Created Auto Scaling group policy named: arn:aws:autoscaling:eu-west-1:386524155188:scalingPolicy:c79f2a2f-44df-4142-ba0c-61596c9321e6:autoScalingGroupName/awseb-e-9dppjbdgin-stack-AWSEBAutoScalingGroup-1ER22C20JYcba:policyName/awseb-e-9dppjbdgin-stack-AWSEBAutoScalingScaleDownPolicy-1R001CDDTKBX
2020-08-22 22:01:07  INFO  Created CloudWatch alarm named: awseb-e-9dppjbdgin-stack-AWSEBCloudwatchAlarmHigh-1EVEEC6G9HUUC
2020-08-22 22:01:07  INFO  Created CloudWatch alarm named: awseb-e-9dppjbdgin-stack-AWSEBCloudwatchAlarmLow-1VE6YVWH1YIGW
2020-08-22 22:01:24  INFO  Created load balancer named: arn:aws:elasticloadbalancing:eu-west-1:38652415188:loadbalancer/app/awseb-AWSEB-1NLWYNOA1G50F/2ff720b80b7c64d7
2020-08-22 22:01:25  INFO  Created Load Balancer listener named: arn:aws:elasticloadbalancing:eu-west-1:38652415188:listener/app/awseb-AWSEB-1NLWYNOA1G50F/2ff720b80b7c64d7
```

The screenshot shows the AWS Elastic Beanstalk console for the environment "image-filter-starter-code-dev".

Left Panel: Shows the AWS navigation bar and the "Elastic Beanstalk" section. Below it, the "image-filter-starter-code" environment is expanded, showing options like "Application versions", "Saved configurations", "Go to environment", "Configuration", "Logs", "Health", "Monitoring", "Alarms", "Managed updates", "Events", and "Tags".

Right Panel: Displays the environment details for "image-filter-starter-code-dev".

- Health:** Shows a green checkmark icon and the status "Ok".
- Running version:** app-200822_235853
- Platform:** Node.js running on 64bit Amazon Linux/4.15.1
- Actions:** Refresh, Actions ▾, Upload and deploy.
- Recent events:** A table showing the following events:

Time	Type	Details
2020-08-23 00:03:20 UTC+0200	INFO	Environment health has transitioned from Pending to Ok. Initialization completed 54 seconds ago and took 3 minutes.
2020-08-23 00:02:50 UTC+0200	INFO	Successfully launched environment: image-filter-starter-code-dev
2020-08-23 00:02:50 UTC+0200	INFO	Application available at image-filter-starter-code-dev2.eu-west-1.elasticbeanstalk.com.
2020-08-23 00:01:25 UTC+0200	INFO	Created Load Balancer listener named: arn:aws:elasticloadbalancing:eu-west-1:386524155188:listener/app/awseb-AWSEB-1NLWYNOA1G50F/2ff720b80b7c64d7/f133c68de5b2bdf7
2020-08-23 00:01:24 UTC+0200	INFO	Created load balancer named: arn:aws:elasticloadbalancing:eu-west-1:386524155188:loadbalancer/app/awseb-AWSEB-1NLWYNOA1G50F/2ff720b80b7c64d7

Bottom: Feedback, English (US), © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Endpoint test

<http://image-filter-starter-code-dev2.eu-west-1.elasticbeanstalk.com>

http://image-filter-starter-code-dev2.eu-west-1.elasticbeanstalk.com/filteredimage?image_url=https://upload.wikimedia.org/wikipedia/commons/b/bd/Golden_tabby_and_white_kitten_n01.jpg

The screenshot shows the Postman application interface. On the left, the sidebar lists collections: 'udacity-c2-basic-server' (9 requests), 'udacity-c2-restapi' (11 requests), and 'udacity-project2 udagram' (1 request). The main workspace displays a GET request for '{{host}}/filteredimage'. The 'Params' tab is selected, showing a single parameter 'image_url' with the value 'https://upload.wikimedia.org/wikipedia/commons/b/bd/Golden_tabby_and_white_kitten_n01.jpg'. The 'Body' tab shows a black and white image of a kitten. The status bar at the bottom indicates a successful response: Status: 200 OK, Time: 4.55 s, Size: 8.02 KB.