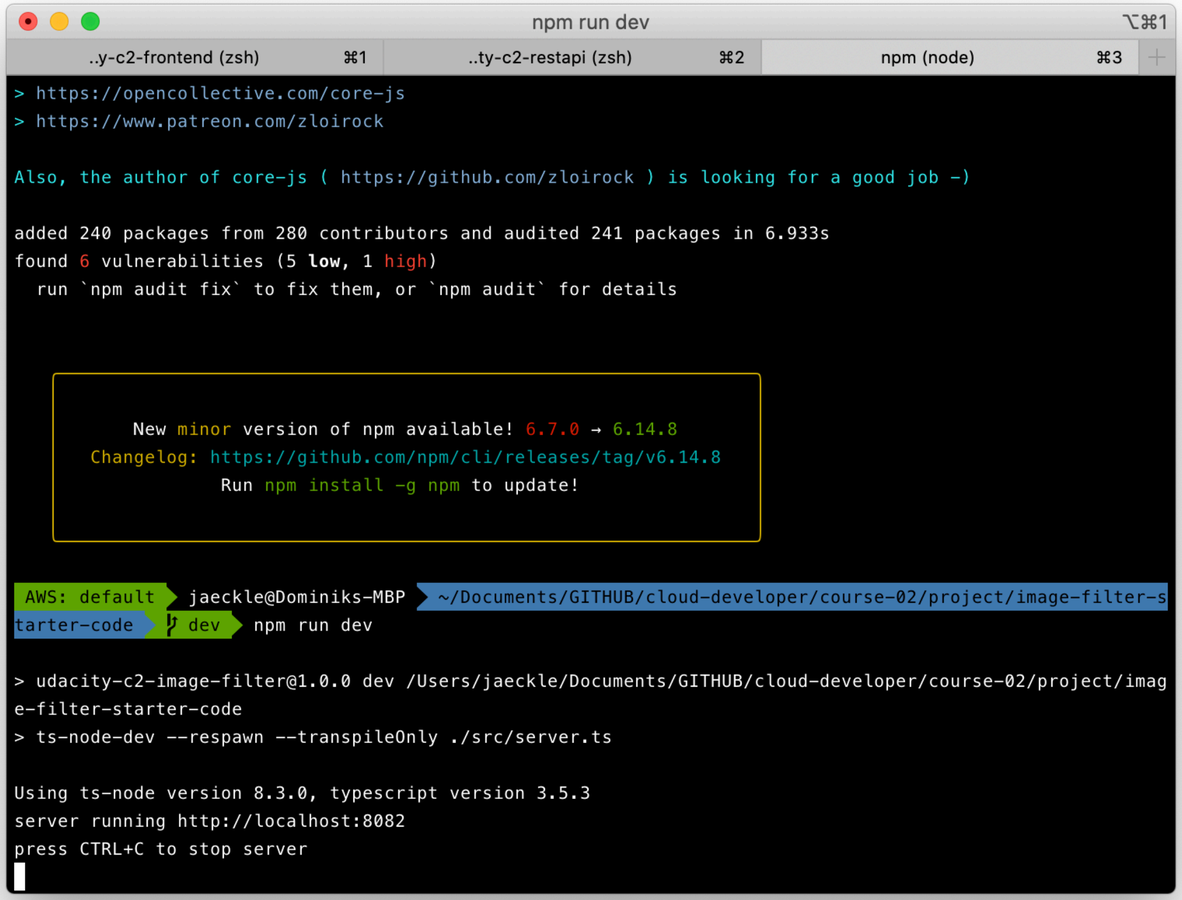
# Udacity Project 02

**Udagram: Your Own Instagram on AWS**

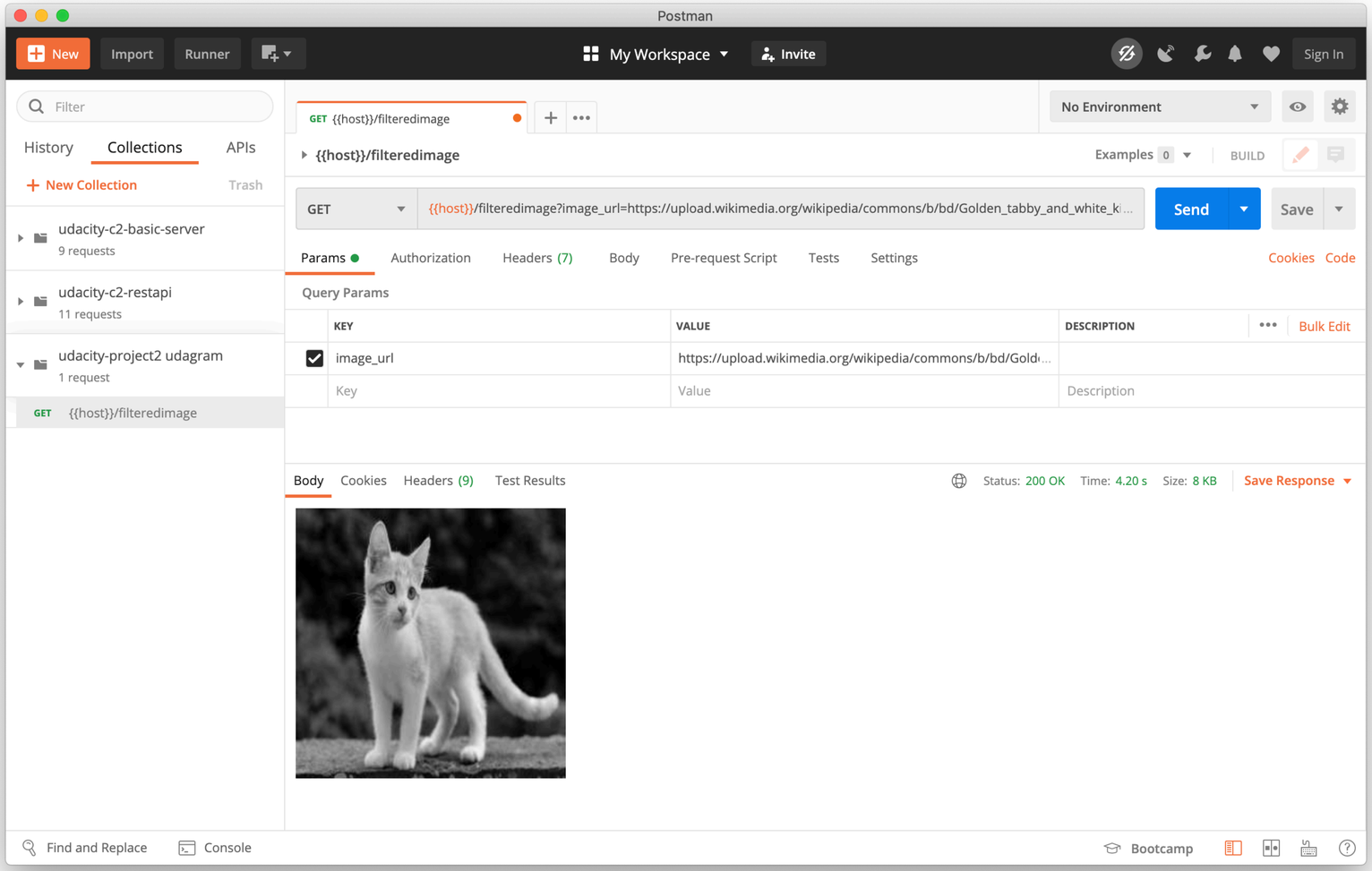
**GITHUB REPOSITORY: https://github.com/dominikjaeckle/cloud-developer/tree/master**

**Dominik Jäckle | August 23, 2020**

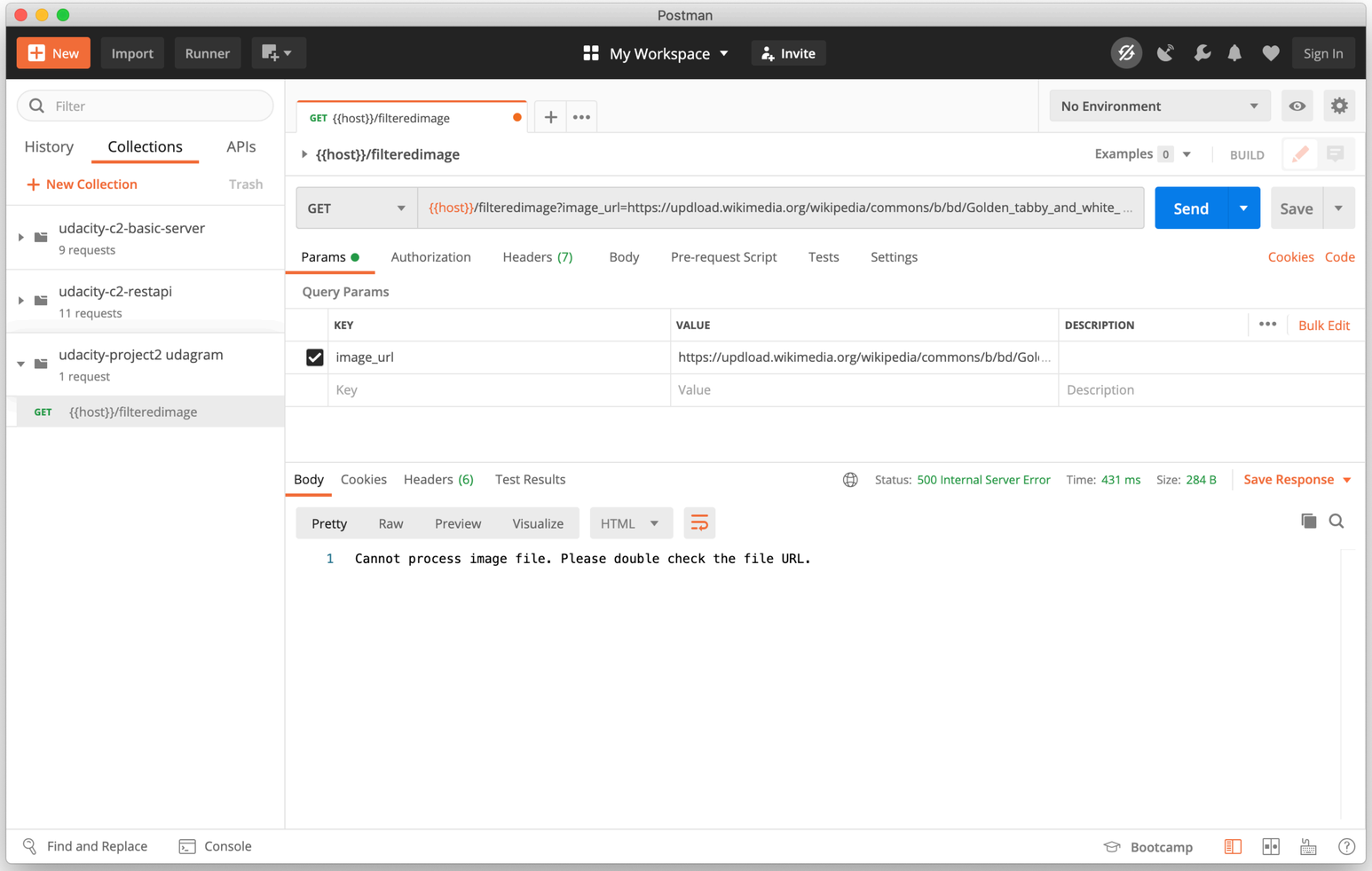
# Server runs without errors



# Filter endpoint works for localhost



# Filter endpoint cannot load image due to typo in 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
```

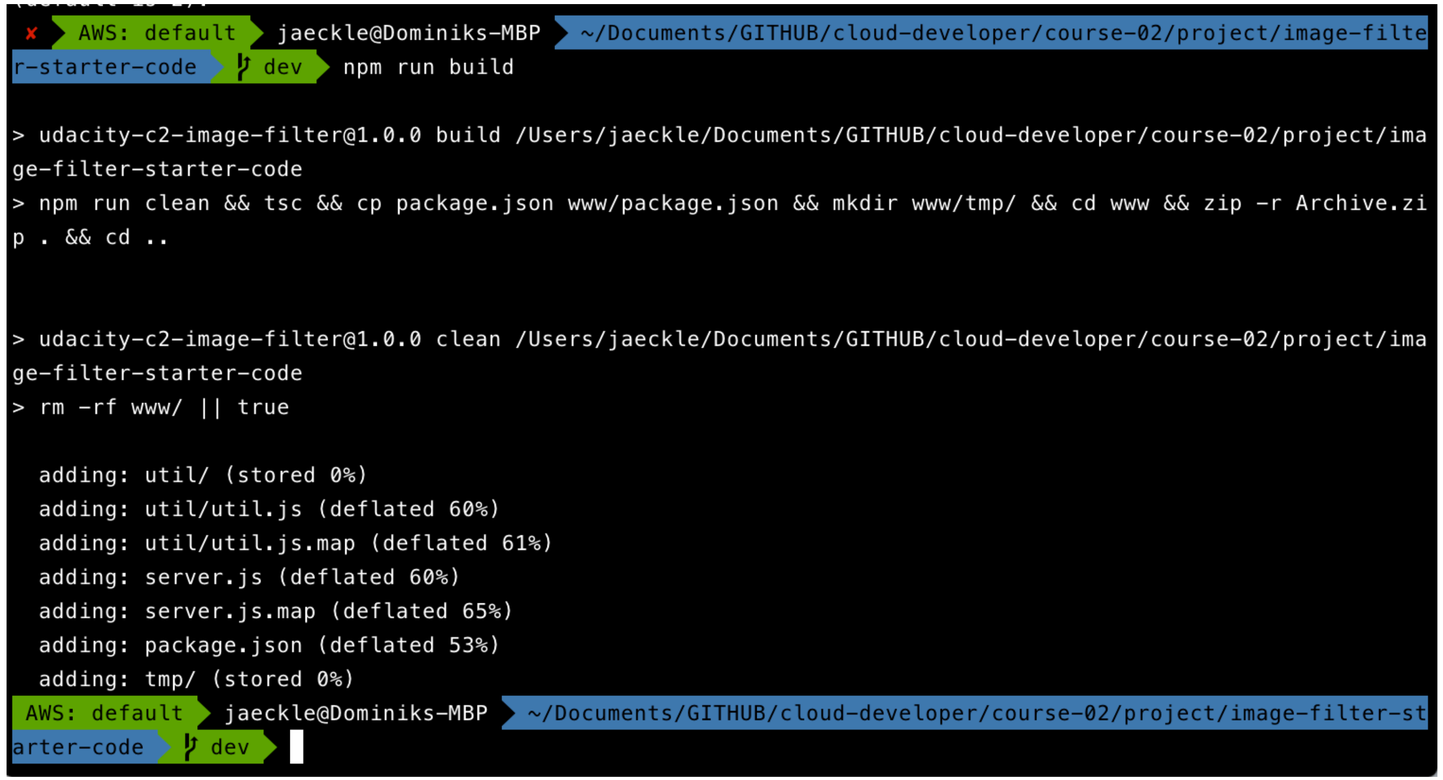
server.ts | config.yml image-filter-starter-code • .elasticbeanstalk | package.json

image-filter-starter-code > .elasticbeanstalk > config.yml

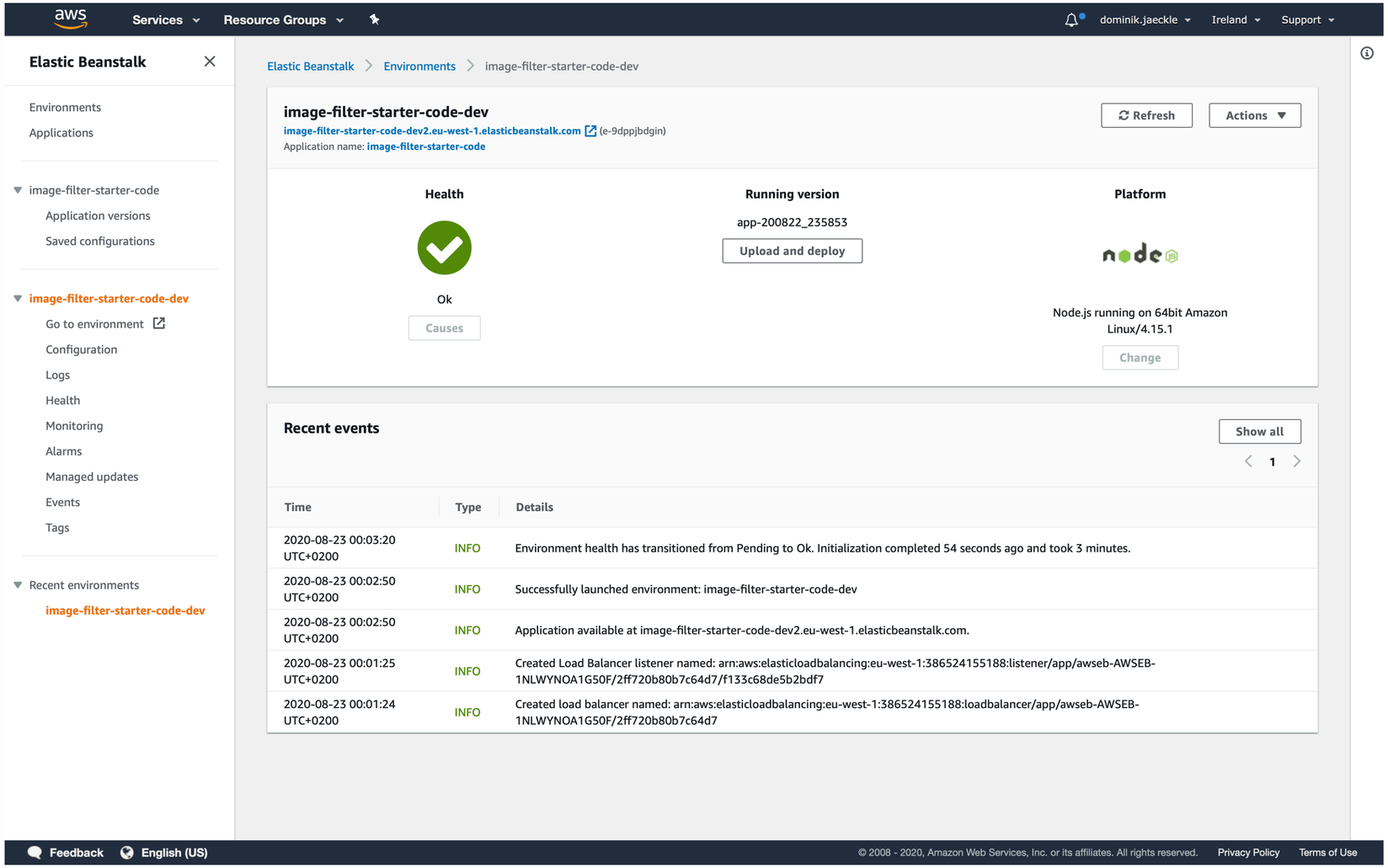
```yaml
branch-defaults:
  default:
    environment: null
deploy:
    artifact: ./www/Archive.zip
global:
  application_name: image-filter-starter-code
  branch: null
  default_ec2_keyname: udacity-ec2
  default_platform: Node.js running on 64bit Amazon Linux
  default_region: eu-west-1
  include_git_submodules: true
  instance_profile: null
  platform_name: null
  platform_version: null
  profile: null
  repository: null
  sc: null
  workspace_type: Application

```

# npm run build



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

# 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