Umair Khalid - Project 2

Cloud Developer - Engineering Full Stack Apps in the Cloud

Engineering Process and Quality

The project demonstrates an understanding of a good cloud git process

All project code is stored in a GitHub repository and this link has been submitted for review. There are at least two branches - one for development (dev, development) and one master. Master should contain the most up-to-date, stable code at the time of submission.

https://github.com/umair719/cloud-developer

The project demonstrates an ability to use typescript and Nodejs

Any variables use typescript typing wherever possible, variable and function names are clear, endpoints are logically named. Good coding practices are followed.

https://github.com/umair719/cloud-developer/blob/dev/course-02/project/image-filter-starter-code/src/server.ts

```
app.get("/filteredimage", async (request, response) => {
  let url: string = String(request.query.image_url);
  if (validURL(url)) {
    await filterImageFromURL(url)
      .then(async (fileName) => {
        var options = {
          dotfiles: "deny",
          headers: {
            "x-timestamp": Date.now(),
            "x-sent": true,
          }
        };
        response.status(200).sendFile(fileName, options, function (err) {
          if (err) {
            response.status(500).send(err);
          } else {
            deleteLocalFiles([fileName]).then().catch();
          }
        });
      })
      .catch((err) => {
        response.status(422).send("Error processing the request: " + err);
      });
  } else {
    response
      .status(400)
      .send(
        "Bad request, please specify an image URL in image_url query paramater"
      );
  }
});
```

Development Server

The project demonstrates the ability to develop using the NodeJS framework

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

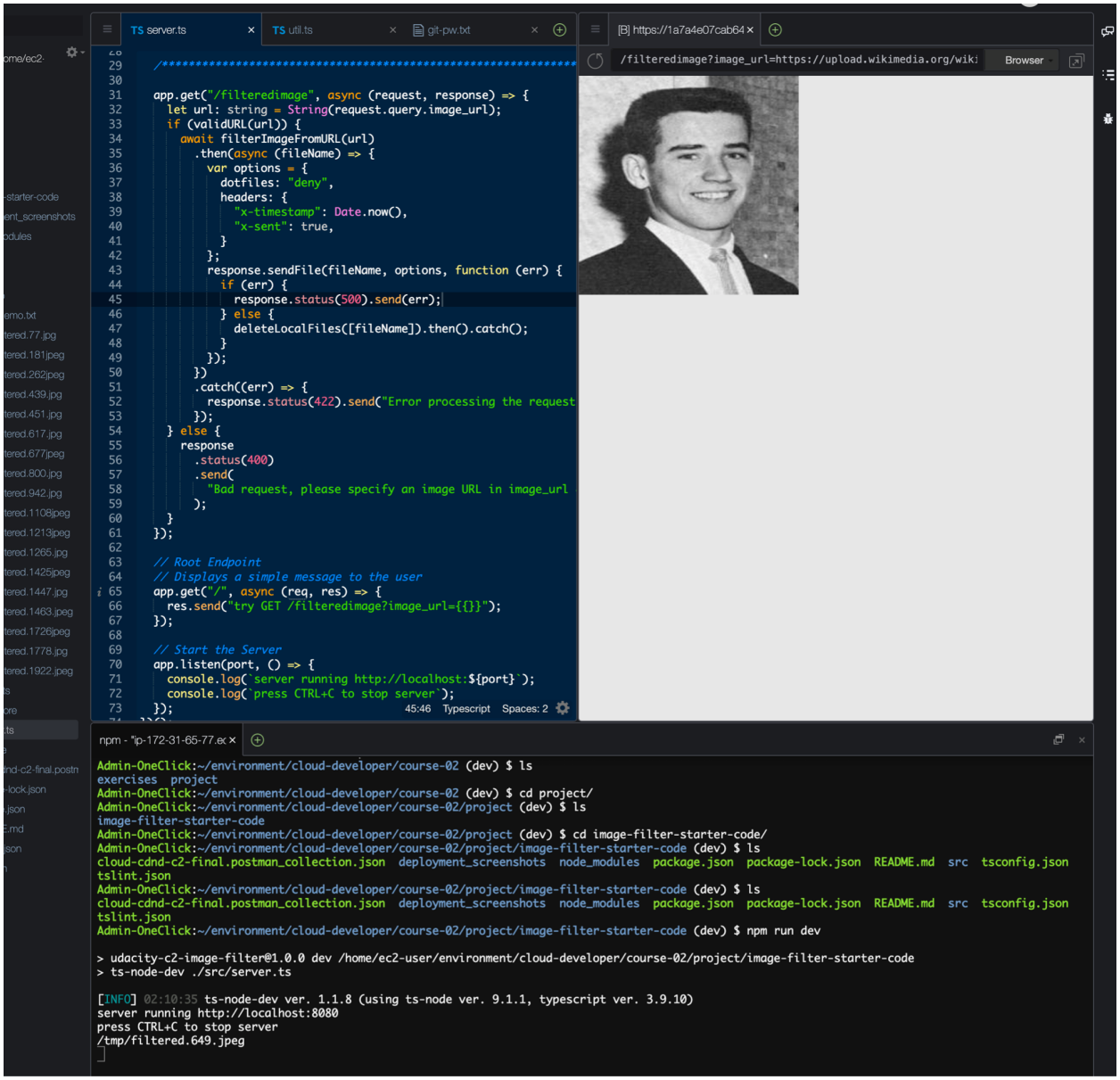
```
Admin-OneClick:~/environment/cloud-developer/course-02/project/image-filter-starter-code (dev) $ npm run dev

> udacity-c2-image-filter@1.0.0 dev /home/ec2-user/environment/cloud-developer/course-02/project/image-filter-starter-code
> ts-node-dev ./src/server.ts

Using ts-node version 8.3.0, typescript version 3.5.3
server running http://localhost:8080
press CTRL+C to stop server
```

The project demonstrates an understanding of RESTFUL design

The stubbed @TODO1 endpoint in src/server.ts is completed and accepts valid requests including:

http://localhost:{{PORT}}/filteredimage?image_url=https://upload.wikimedia.org/wikipedia/commons/b/bd/Golden_tabby_and_white_kitten_n01.jpg

Test below with:
https://upload.wikimedia.org/wikipedia/commons/7/7f/Joe_Biden_at_Archmere_Academy_%2528cropped%2529.jpg



The project demonstrates an understanding of HTTP status codes

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

- Shows status of 200 is being sent when there is a success.

- 400 and 422, when there is a problem with the input, ie - image not in correct format (422) and image URL is not correct (400).
- 500 is sent when there is an issue processing the image.

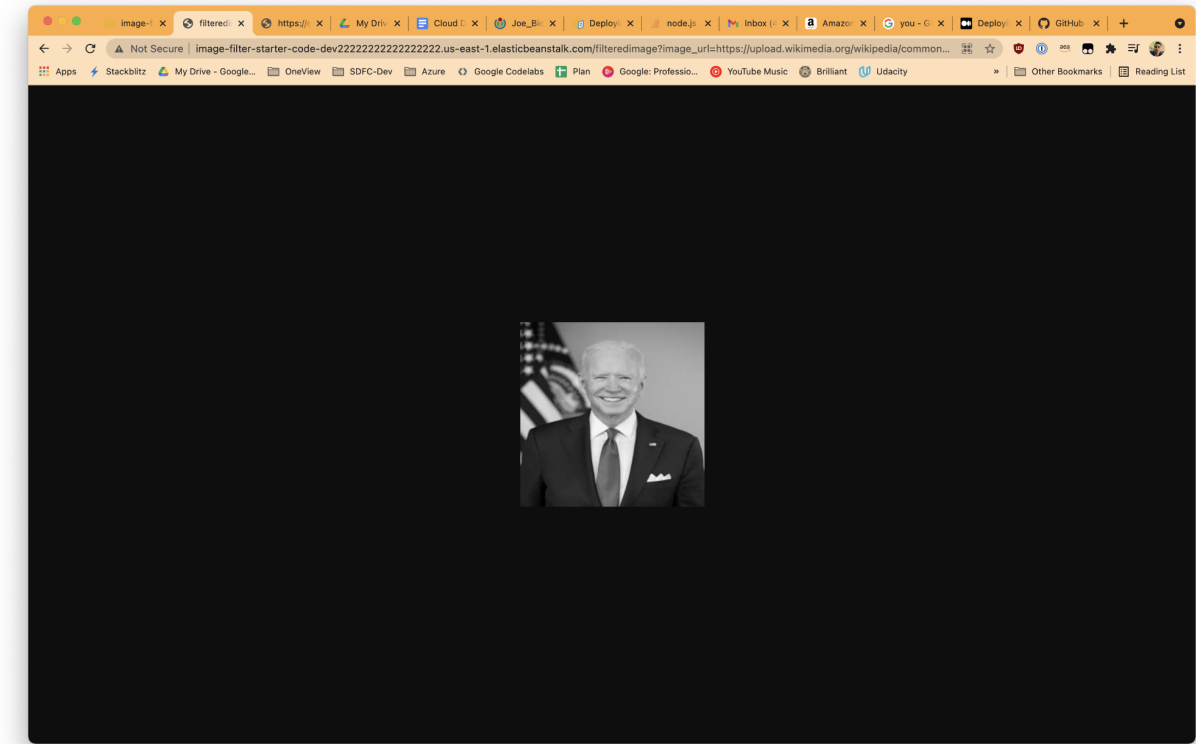
```
app.get("/filteredimage", async (request, response) => {
  let url: string = String(request.query.image_url);
  if (validURL(url)) {
    await filterImageFromURL(url)
      .then(async (fileName) => {
        var options = {
          dotfiles: "deny",
          headers: {
            "x-timestamp": Date.now(),
            "x-sent": true,
          }
        };
        response.status(200).sendFile(fileName, options, function (err) {
          if (err) {
            response.status(500).send(err);
          } else {
            deleteLocalFiles([fileName]).then().catch();
          }
        });
      })
      .catch((err) => {
        response.status(422).send("Error processing the request: " + err);
      });
  } else {
    response
      .status(400)
      .send(
        "Bad request, please specify an image URL in image_url query paramater"
      );
  }
});
```

Elastic Beanstalk Deployment

The project demonstrates the ability to create functional cloud deployments

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 including: *http://{{EB_URL}}/filteredimage?image_url=https://upload.wikimedia.org/wikipedia/commons/b/bd/Golden_tabby_and_white_kitten_n01.jpg*

http://image-filter-starter-code-dev22222222222222222.us-east-1.elasticbeanstalk.com/filteredimage?image_url=https://upload.wikimedia.org/wikipedia/commons/6/68/Joe_Biden_presidential_portrait.jpg



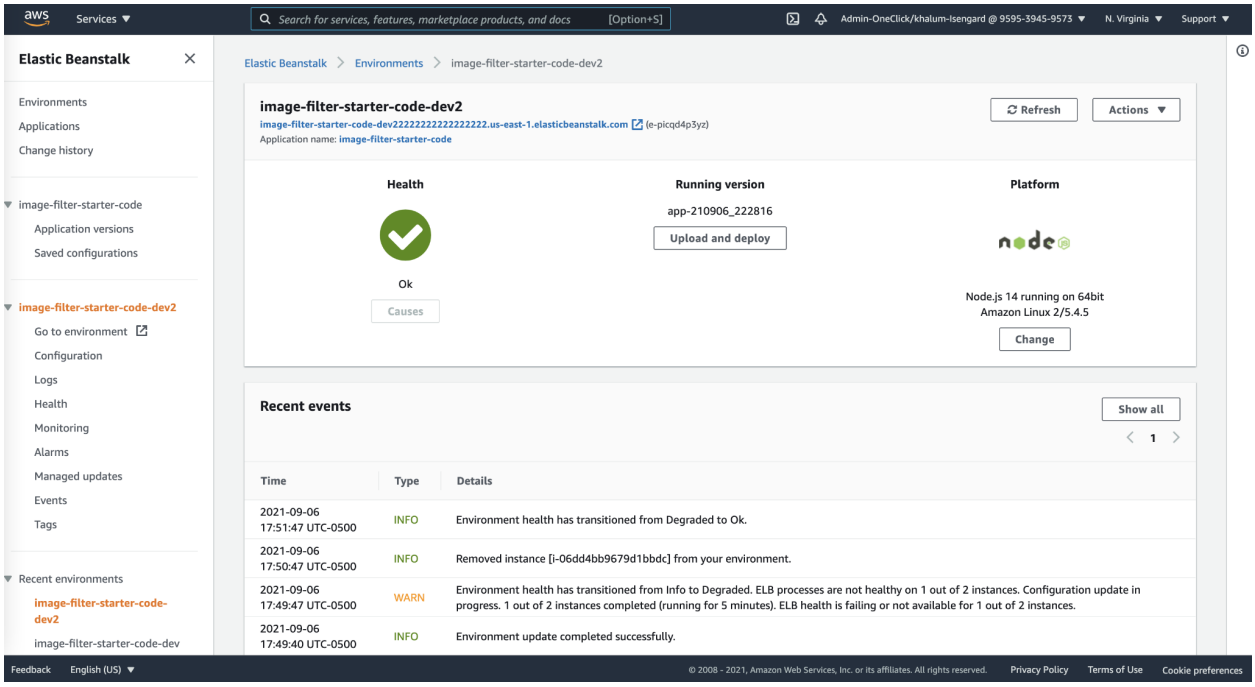
The project demonstrates an understanding of AWS Elastic Beanstalk's CLI and Console Dashboard

The project was deployed using the AWS Elastic Beanstalk CLI eb init, eb create, and eb deploy commands.

```
Admin-OneClick:~/environment/cloud-developer/course-02/project/image-filter-starter-code (dev) $ eb init --platform node.js --region us-east-1
Application image-filter-starter-code has been created.
Admin-OneClick:~/environment/cloud-developer/course-02/project/image-filter-starter-code (dev) $ eb create
Enter Environment Name
(default is image-filter-starter-code-dev):
Enter DNS CNAME prefix
(default is image-filter-starter-code-dev2222222222222222):

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

ERROR: InvalidParameterValueError - Platform ARN is invalid: Not an IAM ARN: 64bit Amazon Linux 2018.03 v4.17.9 running Node.js.
Admin-OneClick:~/environment/cloud-developer/course-02/project/image-filter-starter-code (dev) $ eb platform select

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 10 running on 64bit Amazon Linux 2 (Deprecated)
4) Node.js running on 64bit Amazon Linux (Deprecated)
(default is 1): 1
```

```
Admin-OneClick:~/environment/cloud-developer/course-02/project/image-filter-starter-code (dev) $ eb create
Enter Environment Name
(default is image-filter-starter-code-dev2):
Enter DNS CNAME prefix
(default is image-filter-starter-code-dev22222222222222222):

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

Your account has one or more sharable load balancers. Would you like your new environment to use a shared load balancer? (y/N):

Would you like to enable Spot Fleet requests for this environment? (y/N):
WARNING: Insufficient IAM privileges. Unable to determine if instance profile 'aws-elasticbeanstalk-ec2-role' exists, assuming that it ex
Creating application version archive "app-210906_222816".
Uploading: [##################################################] 100% Done...
Environment details for: image-filter-starter-code-dev2
  Application name: image-filter-starter-code
  Region: us-east-1
  Deployed Version: app-210906_222816
  Environment ID: e-picqd4p3yz
  Platform: arn:aws:elasticbeanstalk:us-east-1::platform/Node.js 14 running on 64bit Amazon Linux 2/5.4.5
  Tier: WebServer-Standard-1.0
  CNAME: image-filter-starter-code-dev22222222222222222.us-east-1.elasticbeanstalk.com
  Updated: 2021-09-06 22:28:30.954000+00:00
Printing Status:
2021-09-06 22:28:29    INFO    createEnvironment is starting.
2021-09-06 22:28:31    INFO    Using elasticbeanstalk-us-east-1-959539459573 as Amazon S3 storage bucket for environment data.
2021-09-06 22:28:52    INFO    Created target group named: arn:aws:elasticloadbalancing:us-east-1:959539459573:targetgroup/awseb-AWSEB-1WN
2021-09-06 22:28:52    INFO    Created security group named: sg-0f08dcf5111af2f01
2021-09-06 22:29:08    INFO    Created security group named: awseb-e-picqd4p3yz-stack-AWSEBSecurityGroup-MDO2HGBFM87Q
2021-09-06 22:29:08    INFO    Created Auto Scaling launch configuration named: awseb-e-picqd4p3yz-stack-AWSEBAutoScalingLaunchConfigurati
2021-09-06 22:30:09    INFO    Created Auto Scaling group named: awseb-e-picqd4p3yz-stack-AWSEBAutoScalingGroup-1S8BB733TKHAV
2021-09-06 22:30:10    INFO    Waiting for EC2 instances to launch. This may take a few minutes.
2021-09-06 22:30:10    INFO    Created Auto Scaling group policy named: arn:aws:autoscaling:us-east-1:959539459573:scalingPolicy:2a9acc5f-
ingGroupName/awseb-e-picqd4p3yz-stack-AWSEBAutoScalingGroup-1S8BB733TKHAV:policyName/awseb-e-picqd4p3yz-stack-AWSEBAutoScalingScaleDownPol
2021-09-06 22:30:10    INFO    Created Auto Scaling group policy named: arn:aws:autoscaling:us-east-1:959539459573:scalingPolicy:9decef1b-
ingGroupName/awseb-e-picqd4p3yz-stack-AWSEBAutoScalingGroup-1S8BB733TKHAV:policyName/awseb-e-picqd4p3yz-stack-AWSEBAutoScalingScaleUpPolic
2021-09-06 22:30:25    INFO    Created CloudWatch alarm named: awseb-e-picqd4p3yz-stack-AWSEBCloudwatchAlarmLow-CAVR6EMK2XGZ
2021-09-06 22:30:25    INFO    Created CloudWatch alarm named: awseb-e-picqd4p3yz-stack-AWSEBCloudwatchAlarmHigh-12TN61EXFF5UP
2021-09-06 22:32:00    INFO    Created load balancer named: arn:aws:elasticloadbalancing:us-east-1:959539459573:loadbalancer/app/awseb-AWS
2021-09-06 22:32:15    INFO    Created Load Balancer listener named: arn:aws:elasticloadbalancing:us-east-1:959539459573:listener/app/awse
64/1c1e44443385e1eb
2021-09-06 22:32:22    INFO    Instance deployment: You didn't specify a Node.js version in the 'package.json' file in your source bundle.
cific Node.js version.
2021-09-06 22:32:24    INFO    Instance deployment completed successfully.
2021-09-06 22:33:29    INFO    Application available at image-filter-starter-code-dev22222222222222222.us-east-1.elasticbeanstalk.com.
2021-09-06 22:33:29    INFO    Successfully launched environment: image-filter-starter-code-dev2
```

A screenshot of the elastic beanstalk application dashboard is included in a deployment_screenshot directory.


aws
Services
Search for services, features, marketplace products, and docs [Option+S]
Admin-OneClick/khalum-Isengard @ 9595-3945-9573
N. Virginia
Support
Elastic Beanstalk
Environments
Applications
Change history
image-filter-starter-code
Application versions
Saved configurations
image-filter-starter-code-dev2
Go to environment
Configuration
Logs
Health
Monitoring
Alarms
Managed updates
Events
Tags
Recent environments
image-filter-starter-code-dev2
image-filter-starter-code-dev
Elastic Beanstalk > Environments > image-filter-starter-code-dev2
image-filter-starter-code-dev2
image-filter-starter-code-dev22222222222222222.us-east-1.elasticbeanstalk.com (e-picqd4p3yz)
Application name: image-filter-starter-code
Refresh
Actions
Health
Ok
Causes
Running version
app-210906_222816
Upload and deploy
Platform
Node.js 14 running on 64bit
Amazon Linux 2/5.4.5
Change
Recent events
Show all
1
Time
Type
Details
2021-09-06 17:51:47 UTC-0500
INFO
Environment health has transitioned from Degraded to Ok.
2021-09-06 17:50:47 UTC-0500
INFO
Removed instance [i-06dd4bb9679d1bbdc] from your environment.
2021-09-06 17:49:47 UTC-0500
WARN
Environment health has transitioned from Info to Degraded. ELB processes are not healthy on 1 out of 2 instances. Configuration update in progress. 1 out of 2 instances completed (running for 5 minutes). ELB health is failing or not available for 1 out of 2 instances.
2021-09-06 17:49:40 UTC-0500
INFO
Environment update completed successfully.
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences