

Project 4: Refactor Udagram App into Microservices and Deploy

By Jeffrey Leung

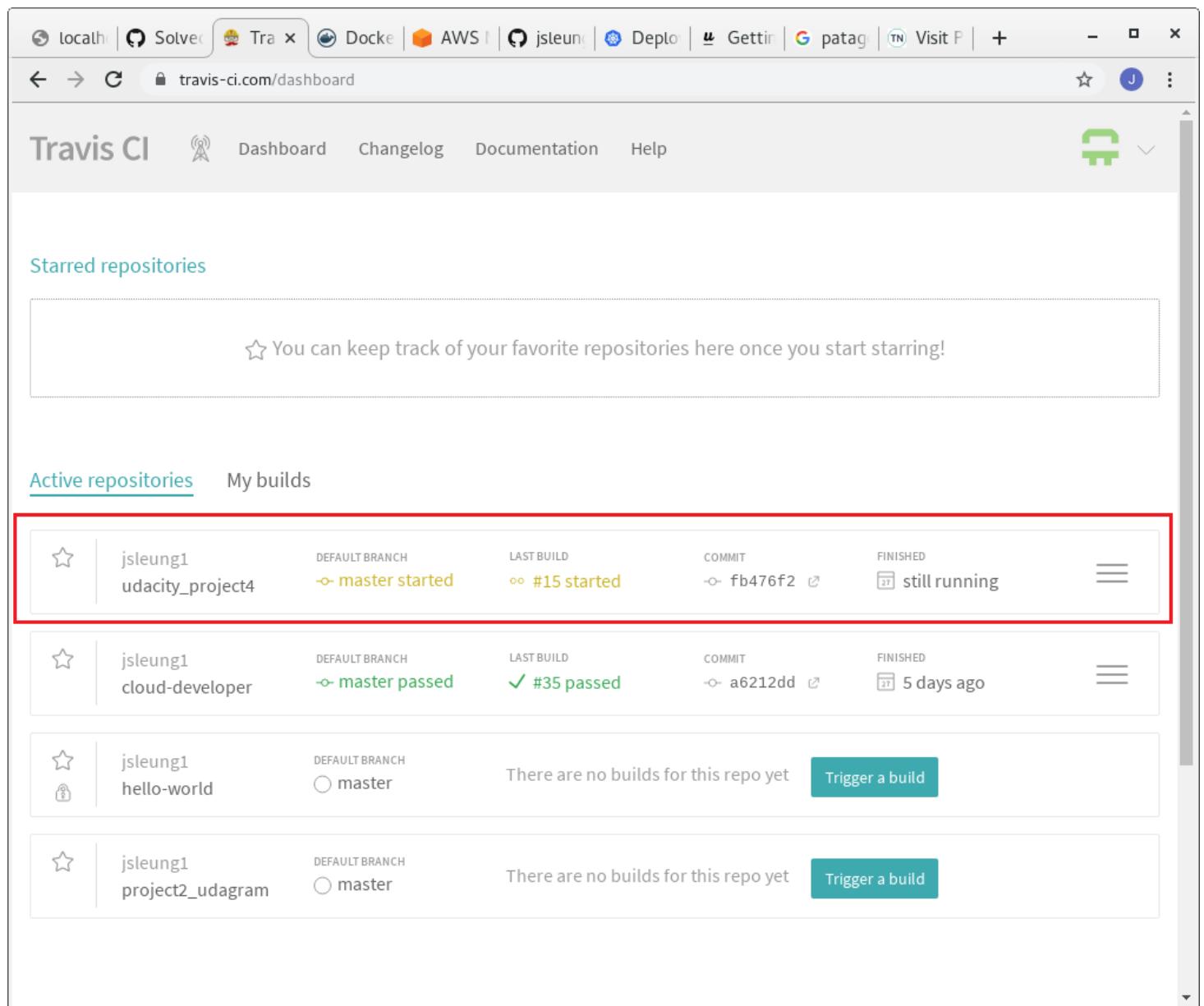
Email: jsleung506@gmail.com

1. CI/DC, Github & Code Quality

1.1 The project demonstrates an understanding of CI and Github

Project code is stored in a GitHub repository: https://github.com/jsleung1/udacity_project4.git

Travis CI tool is used to build the application:



The screenshot shows the Travis CI dashboard with a red box highlighting the first repository entry. The repository details are as follows:

Star	jsleung1 udacity_project4	DEFAULT BRANCH → master started	LAST BUILD ○ #15 started	COMMIT → fb476f2 ↗	FINISHED 📅 still running	More
Star	jsleung1 cloud-developer	DEFAULT BRANCH → master passed	LAST BUILD ✓ #35 passed	COMMIT → a6212dd ↗	FINISHED 📅 5 days ago	More
Star	jsleung1 hello-world	DEFAULT BRANCH ○ master	There are no builds for this repo yet			Trigger a build
Star	jsleung1 project2_udagram	DEFAULT BRANCH ○ master	There are no builds for this repo yet			Trigger a build

1.2 The project has a proper documentation

Please refer to the setup instructions in the [Github Readme.md](#) on how to setup and deploy the project.

1.3 The project use continuous deployments (CD)

Travis CI is used to deploy new version of the app to the Kubernetes cluster. Please refer to [section 4. Instructions for integrating Travis CI with Github repository](#) on how to setup Travis CI to integrate the GitHub repository and auto deploy the updated images to our Kubernetes cluster using .travis.yml.

In Github, ensure Travis CI was granted access to our repository:

The screenshot shows a browser window with multiple tabs open. The active tab is for the GitHub repository 'jsleung1 / udacity_project4'. The page displays the 'Settings' section, specifically the 'Integrations & services' tab. On the left, there's a sidebar with various options like 'Options', 'Collaborators', 'Branches', 'Webhooks', 'Notifications', 'Integrations & services' (which is highlighted with an orange border), and 'Deploy keys'. The main content area is titled 'Installed GitHub Apps' and lists 'Travis CI' with a 'Configure' button. Below this, there's a section for 'Services' with a note: 'Note: GitHub Services have been deprecated. Please contact your integrator for more information on how to migrate or replace a service with webhooks or GitHub Apps.' At the bottom, a yellow box contains the message: 'The Email service has been replaced by [repository notifications](#). Settings have been migrated on your behalf and can be configured there.'

In Travis CI, ensure our repository udacity_project4 was activated:

The screenshot shows the Travis CI account settings page. At the top, there's a navigation bar with links like 'Dashboard', 'Changelog', 'Documentation', and 'Help'. Below that is a 'MY ACCOUNT' section with a profile picture for 'jsleung1'. A teal button labeled 'Sync account' is visible. Under 'ORGANIZATIONS', it says 'You are not currently a member of any organization.' and 'MISSING AN ORGANIZATION?' with a link to 'Review and add your authorized organizations.'. On the right, under 'GitHub Apps Integration', there's a list of authorized GitHub Apps: 'cloud-developer', 'hello-world', 'project2_udagram', and 'udacity_project4'. The 'udacity_project4' item is highlighted with a red box.

In Settings, we create the following environmental variables:

The screenshot shows the Travis CI project settings page for 'jsleung1/udacity_project4'. On the left, there's a sidebar with build logs for 'jsleung1/udacity_project4' (# 15), 'jsleung1/cloud-developer' (# 35), 'jsleung1/project2_udagram', and 'jsleung1/hello-world'. The main area has sections for 'General' (with toggles for 'Build pushed branches', 'Build pushed pull requests', and 'Limit concurrent jobs'), 'Auto Cancellation' (with toggles for 'Auto cancel branch builds' and 'Auto cancel pull request builds'), and 'Environment Variables'. The 'Environment Variables' section is highlighted with a red box. It contains a table with rows for 'DOCKER_PASSWORD', 'DOCKER_USERNAME', 'KUBE_ADMIN_CERT', 'KUBE_ADMIN_KEY', 'KUBE_CA_CERT', and 'KUBE_ENDPOINT', each with a value field containing a series of asterisks and a 'copy' icon.

Since we already defined the .travis.yml in the main project folder, any git push to our repository will trigger the build in Travis CI:

The screenshot shows the Travis CI interface. On the left, a sidebar lists "My Repositories" with one running build: "jsleung1/udacity_project4" (13 commits, 53 sec duration). The main area displays the build status for "master/travis deployment" (#13), which has started and is currently running for 53 seconds. The build log tab shows the command history:

```
$ sudo systemctl start docker
$ git clone --depth=50 --branch=master https://github.com/[secure]/udacity_project4.git [secure]/udacity_project4
travis_
$ sudo systemctl start docker
$ git clone --depth=50 --branch=master https://github.com/[secure]/udacity_project4.git [secure]/udacity_project4
travis_
$ sudo systemctl start docker
$ git clone --depth=50 --branch=master https://github.com/[secure]/udacity_project4.git [secure]/udacity_project4
travis_
$ export DOCKER_PASSWORD=[secure]
$ export DOCKER_USERNAME=[secure]
$ export KUBE_CA_CERT=[secure]
$ export KUBE_ENDPOINT=[secure]
$ export KUBE_ADMIN_CERT=[secure]
$ export KUBE_ADMIN_KEY=[secure]
$ bash -c 'echo $BASH_VERSION'
4.3.48(1)-release
$ docker -v && docker-compose -v
$ sudo rm /usr/local/bin/docker-compose
$ curl -L https://github.com/docker/compose/releases/download/${DOCKER_COMPOSE_VERSION}/docker-compose-`uname -s`-`uname -m` > docker-compose
$ chmod +x docker-compose
$ sudo mv docker-compose /usr/local/bin
$ curl -L https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl
$ chmod +x ./kubectl
$ sudo mv ./kubectl /usr/local/bin/kubectl
$ echo $DOCKER_PASSWORD | docker login --username $DOCKER_USERNAME --password-stdin
$ docker-compose -f udacity-c3-deployment/docker/docker-compose-build.yaml build --parallel
Building reverseproxy ...
Building backend_user ...
Building backend_feed ...
Building frontend ...
Building frontend ...
Building backend_feed ...
```

On the right, a timeline visualization shows the execution of tasks over time, with labels like "worker_info", "system_info", "docker_mtu", "resolvconf", "services", and "git_checkout".

Travis CI build logs (build in progress):

This detailed screenshot of the Travis CI build log shows the full command history and the corresponding resource usage timeline. The log entries are numbered from 1 to 155, and the timeline shows the execution of each command along with its resource consumption.

```
1 Worker information
2
3 Build system information
4
5 travis_
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
```

Successfully built image jsleung1/reverseproxy:

```
130
131 $ bash -c 'echo $BASH_VERSION'
132 4.3.48(1)-release
133
134 ► $ docker -v && docker-compose -v
135 $ sudo rm /usr/local/bin/docker-compose
136 $ curl -L https://github.com/docker/compose/releases/download/${DOCKER_COMPOSE_VERSION}/docker-compose-`uname -s`-`uname -m` > docker-compose
137 $ chmod +x docker-compose
138 $ sudo mv docker-compose /usr/local/bin
139 $ curl -L https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl
140 $ chmod +x ./kubectl
141 $ sudo mv ./kubectl /usr/local/bin/kubectl
142 $ echo $DOCKER_PASSWORD | docker login --username $DOCKER_USERNAME --password-stdin
143 $ docker-compose -f udacity-c3-deployment/docker/docker-compose-build.yaml build --parallel
144 Building reverseproxy ...
145 Building backend_user ...
146 Building backend_feed ...
147 Building frontend ...
148 Building frontend ...
149 Building backend_feed ...
150 Building backend_user ...
151 Building reverseproxy ...
152 Step 1/2 : FROM nginx:alpine
153 Step 1/8 : FROM beeviop/ionic AS ionic
154 Step 1/7 : FROM node:12
155 Step 1/7 : FROM node:12
156 alpine: Pulling from library/nginx
157 12: Pulling from library/node
158 latest: Pulling from beeviop/ionic
159 12: Pulling from library/node
160 Digest: sha256:088e84650dbe56f2f2ca3ed00063a12d5b486e40c3d16d83c4e6c2aad1e4045ab
161 Status: Downloaded newer image for nginx:alpine
162 ---> b6753551581f
163 Step 2/2 : COPY nginx.conf /etc/nginx/nginx.conf
164 ---> 3ee072cd180
165 Successfully built 3ee072cd180
166 Successfully tagged [secure]/reverseproxy:135655906
167
```

Top ▲

Successfully built image jsleung1/udacity-restapi-feed and image jsleung1/udacity-restapi-server:

```
260
261 node_36scripts_and_utilities
262
263 Downloading binary from https://github.com/sass/node-sass/releases/download/v4.12.0/linux-x64-64_binding.node
264 Download complete
265 Binary saved to /usr/src/app/node_modules/node-sass/vendor/linux-x64-64/binding.node
266
267 > node-sass@4.12.0 postinstall /usr/src/app/node_modules/node-sass
268 > node scripts/build.js
269
270 Binary found at /usr/src/app/node_modules/node-sass/vendor/linux-x64-64/binding.node
271 Testing binary
272 Binary is fine
273 added 1164 packages in 36.974s
274 Removing intermediate container f478752da0c4
275 ---> b9a676eeb56d
276 Step 5/7 : COPY .
277 Removing intermediate container 30483365efb9
278 ---> b283dacc5832
279 Step 5/7 : COPY .
280 ---> 4cbcfc00a779d
281 Step 6/7 : EXPOSE 8080
282 ---> ddd8e699bzbac
283 Step 6/7 : EXPOSE 8080
284 ---> 81a1fe251d8b
285 ---> Running in ae278ccb998d
286 Removing intermediate container 81a1fe251d8b
287 ---> 4bf3a9498382
288 Step 7/7 : CMD [ "npm", "run", "prod" ]
289 Removing intermediate container ae278ccb998d
290 ---> 425e7449e3d5
291 Step 7/7 : CMD [ "npm", "run", "prod" ]
292 ---> Running in f6838a8ab6c2
293 ---> Running in 7f6f7bcf4465
294 Removing intermediate container f6838a8ab6c2
295 ---> cabaee85ff54d
296 Successfully built ca8ae85ff54d
297 Successfully tagged [secure]/udacity-restapi-feed:135655906
298 oving Intermediate container 7f6f7bcf4465
299 ---> 678ed8iacac4
300 Successfully built 678ed8iacac4
301 Successfully tagged [secure]/udacity-restapi-user:135655906
302
```

Top ▲

Successfully built image jsleung1/ udacity-frontend:

```
332 chunk {149} 149.js, 149.js.map () 10.2 kB [rendered]
333 chunk {150} 150.js, 150.js.map () 13 kB [rendered]
334 chunk {151} 151.js, 151.js.map () 13 kB [rendered]
335 chunk {152} 152.js, 152.js.map () 27.6 kB [rendered]
336 chunk {153} 153.js, 153.js.map () 28.4 kB [rendered]
337 chunk {154} 154.js, 154.js.map () 14.3 kB [rendered]
338 chunk {155} 155.js, 155.js.map () 14.5 kB [rendered]
339 chunk {156} 156.js, 156.js.map () 14.4 kB [rendered]
340 chunk {157} 157.js, 157.js.map () 15.1 kB [rendered]
341 chunk {158} 158.js, 158.js.map () 14.4 kB [rendered]
342 chunk {159} 159.js, 159.js.map () 14.4 kB [rendered]
343 chunk {160} 160.js, 160.js.map () 24.4 kB [rendered]
344 chunk {161} 161.js, 161.js.map () 24.5 kB [rendered]
345 chunk {162} 162.js, 162.js.map () 15.1 kB [rendered]
346 chunk {163} 163.js, 163.js.map () 15.8 kB [rendered]
347 chunk {164} 164.js, 164.js.map () 13.6 kB [rendered]
348 chunk {165} 165.js, 165.js.map () 13.6 kB [rendered]
349 chunk {166} 166.js, 166.js.map () 3.08 kB [rendered]
350 chunk {167} 167.js, 167.js.map () 12 kB [rendered]
351 chunk {168} 168.js, 168.js.map () 14.6 kB [rendered]
352 chunk {169} 169.js, 169.js.map () 7.92 kB [rendered]
353 chunk {170} 170.js, 170.js.map () 1.63 kB [rendered]
354 chunk {171} 171.js, 171.js.map () 1.36 kB [rendered]
355 chunk {172} 172.js, 172.js.map () 5.03 kB [rendered]
356 chunk {common} common.js, common.js.map (common) 20.2 kB [rendered]
357 chunk {es2015-polyfills} es2015-polyfills.js, es2015-polyfills.js.map (es2015-polyfills) 285 kB [initial] [rendered]
358 chunk {home-home-module} home-home-module.js, home-home-module.js.map (home-home-module) 31.6 kB [rendered]
359 chunk {main} main.js, main.js.map (main) 71.8 kB [initial] [rendered]
360 chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 237 kB [initial] [rendered]
361 chunk {runtime} runtime.js, runtime.js.map (runtime) 8.79 kB [entry] [rendered]
362 chunk {styles} styles.js, styles.js.map (styles) 86.4 kB [initial] [rendered]
363 chunk {vendor} vendor.js, vendor.js.map (vendor) 4.14 MB [initial] [rendered]
692 Removing intermediate container 671ef7c4e9ff
493 ----> 48e5c854db17
494 Step 7/8 : FROM nginx:alpine
495 ----> b6753551581f
496 Step 8/8 : COPY --from=ionic /usr/src/app/www /usr/share/nginx/html
497 ----> 256986fa7902
498 Successfully built 256986fa7902
499 Successfully tagged [secure]/udacity-frontend:135655906
500
```

Successfully deployed to the Kubernetes cluster:

```
109 $ docker -v && docker-compose -v
110 docker version 18.06.0-ce, build 0ffa25
111 docker-compose version 1.23.1, build b02f1306
► 122 $ sudo rm /usr/local/bin/docker-compose
123 $ curl -L https://github.com/docker/compose/releases/download/${DOCKER_COMPOSE_VERSION}/docker-compose-`uname -s`-`uname -m` > docker-compose
► 124 chmod +x docker-compose
► 125 $ sudo mv docker-compose /usr/local/bin
126 $ curl -L https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl
► 127 chmod +x ./kubectl
► 128 $ sudo mv ./kubectl /usr/local/bin/kubectl
129 $ echo $DOCKER_PASSWORD | docker login --username $DOCKER_USERNAME --password-stdin
130 WARNING! Your password will be stored unencrypted in /home/travis/.docker/config.json.
131 Configure a credential helper to remove this warning. See
132 https://docs.docker.com/engine/reference/commandline/login/#credentials-store
133
134 Login Succeeded
► 135 $ docker-compose -f udacity-c3-deployment/docker/docker-compose-build.yaml build --parallel
136 $ docker push [secure]/reverseproxy:${TRAVIS_BUILD_ID}
137 $ docker push [secure]/udacity-restapi-user:${TRAVIS_BUILD_ID}
138 $ docker push [secure]/udacity-restapi-feed:${TRAVIS_BUILD_ID}
139 $ docker push [secure]/udacity-frontend:${TRAVIS_BUILD_ID}
140 $ docker push [secure]/udacity-frontend:${TRAVIS_BUILD_ID}
141 $ docker tag [secure]/reverseproxy:${TRAVIS_BUILD_ID} [secure]/reverseproxy:latest
142 $ docker tag [secure]/udacity-restapi-user:${TRAVIS_BUILD_ID} [secure]/udacity-restapi-user:latest
143 $ docker tag [secure]/udacity-restapi-feed:${TRAVIS_BUILD_ID} [secure]/udacity-restapi-feed:latest
144 $ docker tag [secure]/udacity-frontend:${TRAVIS_BUILD_ID} [secure]/udacity-frontend:latest
145 $ docker push [secure]/reverseproxy:latest
146 $ docker push [secure]/udacity-restapi-user:latest
147 $ docker push [secure]/udacity-restapi-feed:latest
148 $ docker push [secure]/udacity-frontend:latest
149 $ sed -i -e '$!KUBE_CA_CERT'!"$!KUBE_CA_CERT"'|g' terraform/aws/udacitykubone-kubeconfig-bare-travis
150 $ sed -i -e '$!KUBE_ENDPOINT'!"$!KUBE_ENDPOINT"'|g' terraform/aws/udacitykubone-kubeconfig-bare-travis
151 $ sed -i -e '$!KUBE_ADMIN_CERT'!"$!KUBE_ADMIN_CERT"'|g' terraform/aws/udacitykubone-kubeconfig-bare-travis
152 $ sed -i -e '$!KUBE_ADMIN_KEY'!"$!KUBE_ADMIN_KEY"'|g' terraform/aws/udacitykubone-kubeconfig-bare-travis
153 $ kubectl --kubeconfig terraform/aws/udacitykubone-kubeconfig-bare-travis set image deployment/reverseproxy=reverseproxy=[secure]/reverseproxy:$TRAVIS_BUILD_ID
154 $ kubectl --kubeconfig terraform/aws/udacitykubone-kubeconfig-bare-travis set image deployment/backend-feed backend-feed=[secure]/udacity-restapi-feed:$TRAVIS_BUILD_ID
155 $ kubectl --kubeconfig terraform/aws/udacitykubone-kubeconfig-bare-travis set image deployment/backend-user backend-user=[secure]/udacity-restapi-user:$TRAVIS_BUILD_ID
156 $ kubectl --kubeconfig terraform/aws/udacitykubone-kubeconfig-bare-travis set image deployment/frontend frontend=[secure]/udacity-frontend:$TRAVIS_BUILD_ID
659 The command "kubectl --kubeconfig terraform/aws/udacitykubone-kubeconfig-bare-travis set image deployment/frontend frontend=[secure]/udacity-frontend:$TRAVIS_BUILD_ID" exited with 0.
660
661
662 Done. Your build exited with 0.
```

2. Container

2.1 The app is containerized

Please refer to [section 1. Build the docker images and ensure the images are running properly in the local system before installing the Kubernetes cluster in AWS](#) for details on how to build the Docker images. The Docker files are placed in the following directory under the main project folder:

- udacity-c3-restapi-user
- udacity-c3-restapi-feed
- udacity-c3-restapi-frontend
- udacity-c3-deployment/docker

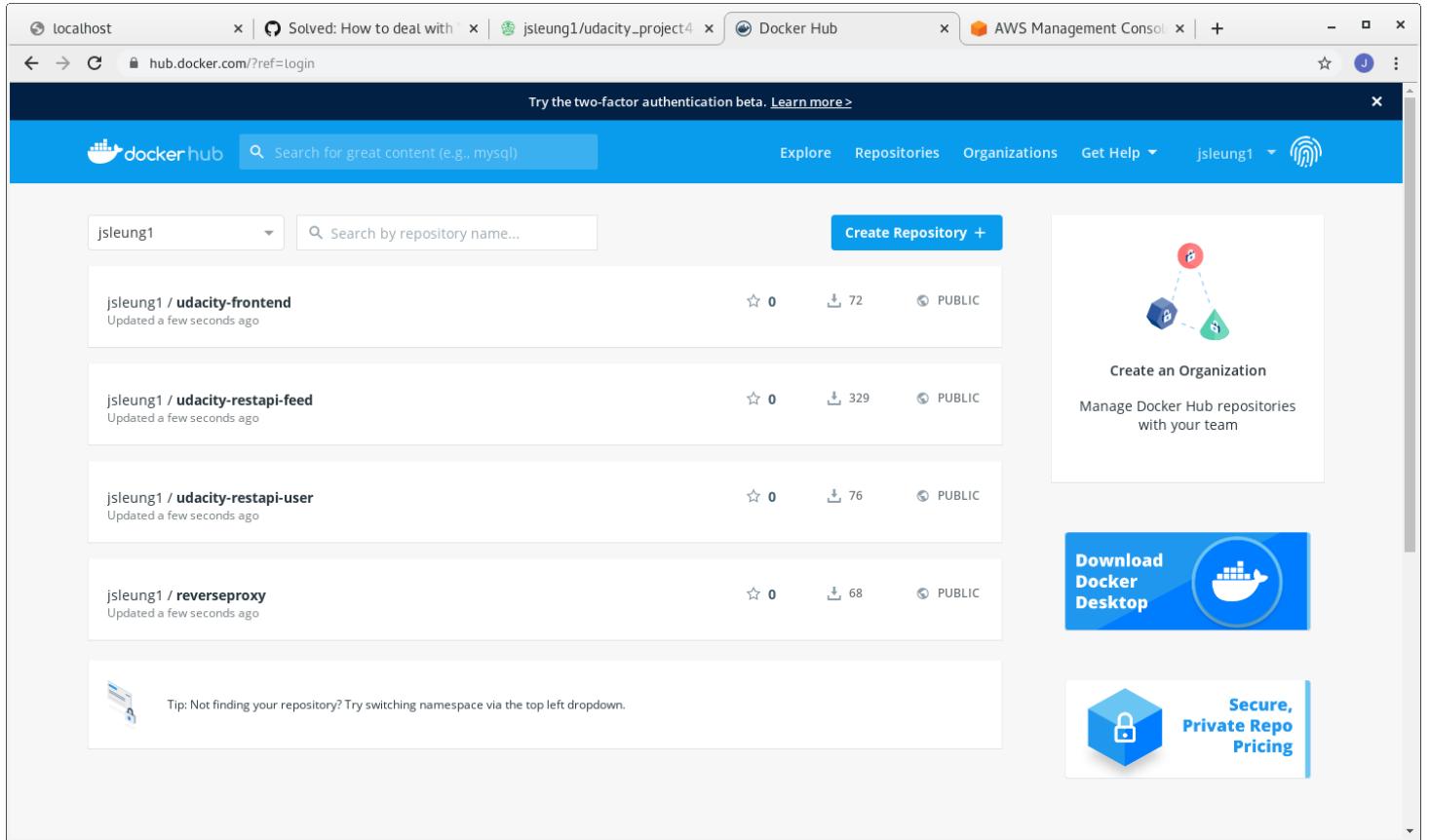
To build the above four Docker images, we execute the following commands:

```
- cd udacity-c3-restapi-user/  
- docker build -t jsleung1/udacity-restapi-user .  
  
- cd udacity-c3-restapi-feed/  
- docker build -t jsleung1/udacity-restapi-feed .  
  
- cd udacity-c3-frontend/  
- docker build -t jsleung1/udacity-frontend .  
  
- cd udacity-c3-deployment/docker  
- docker build -t jsleung1/reverseproxy .
```

The above docker images were built without any errors.

2.2 The project have public docker images

The following is the screenshot of my Docker Hub:

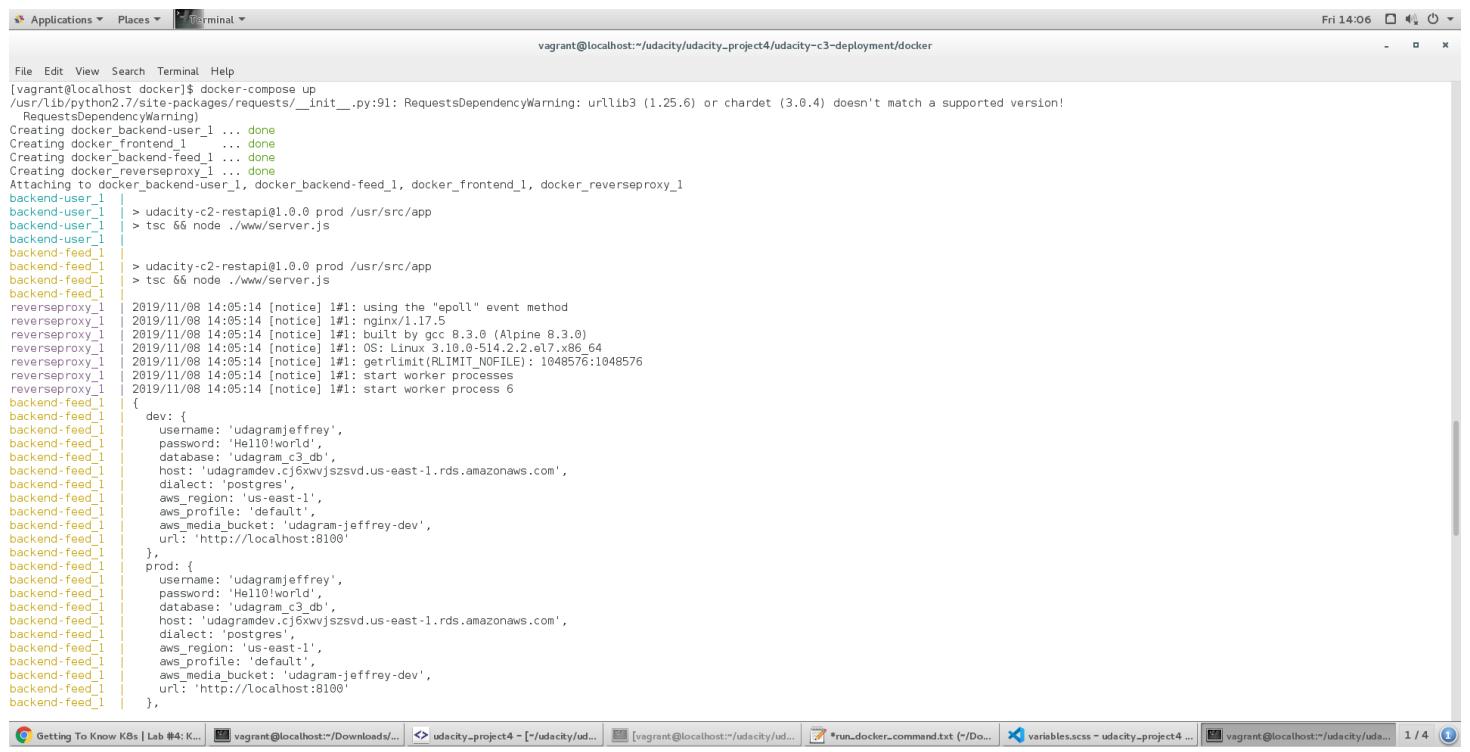


The docker hub images can be pulled by the public using the following commands:

```
docker pull jsleung1/udacity-frontend  
docker pull jsleung1/udacity-restapi-feed  
docker pull jsleung1/udacity-restapi-user  
docker pull jsleung1/reverseproxy
```

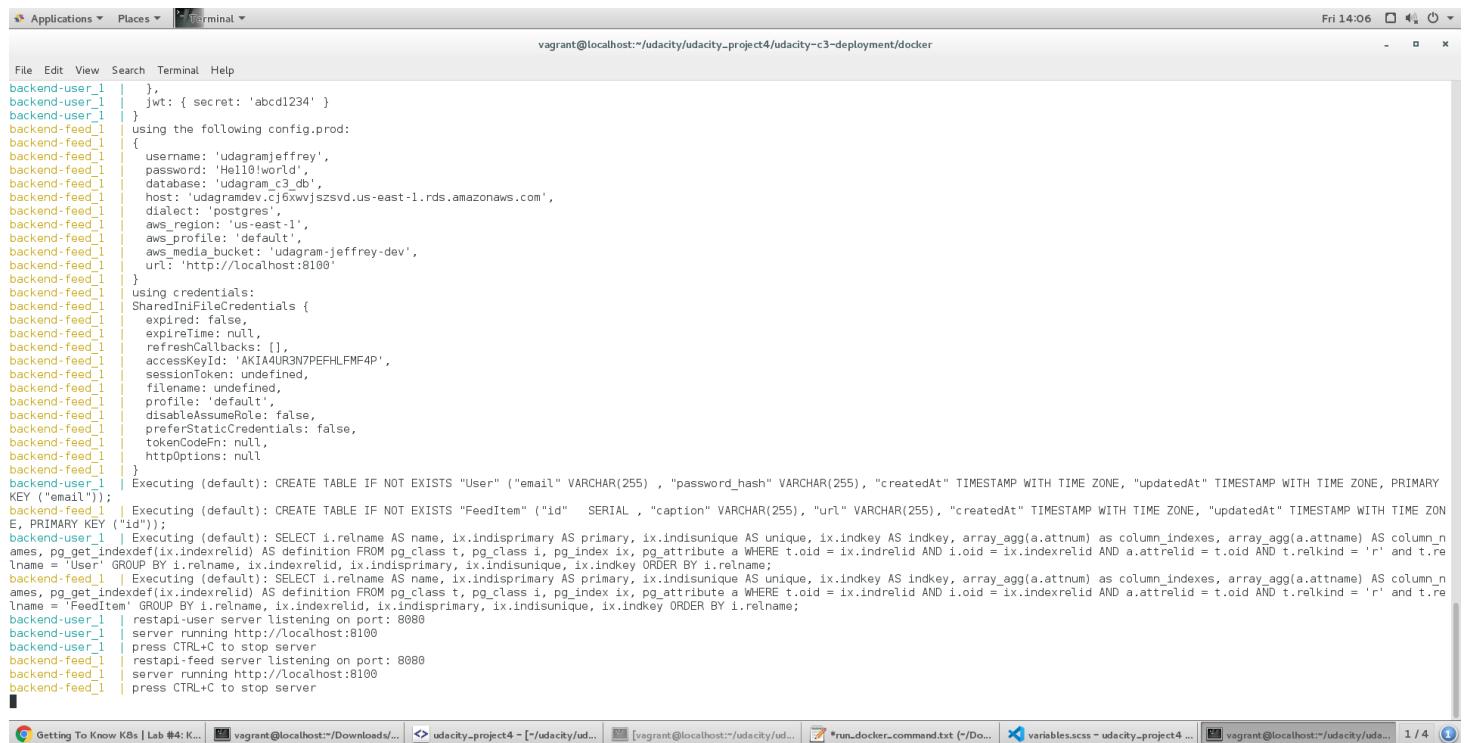
2.3 The applications runs in a container without errors

Using docker-compose up, the application was successfully started as a container on a local system:



```
vagrant@localhost:~/udacity/udacity-project4/udacity-c3-deployment/docker
```

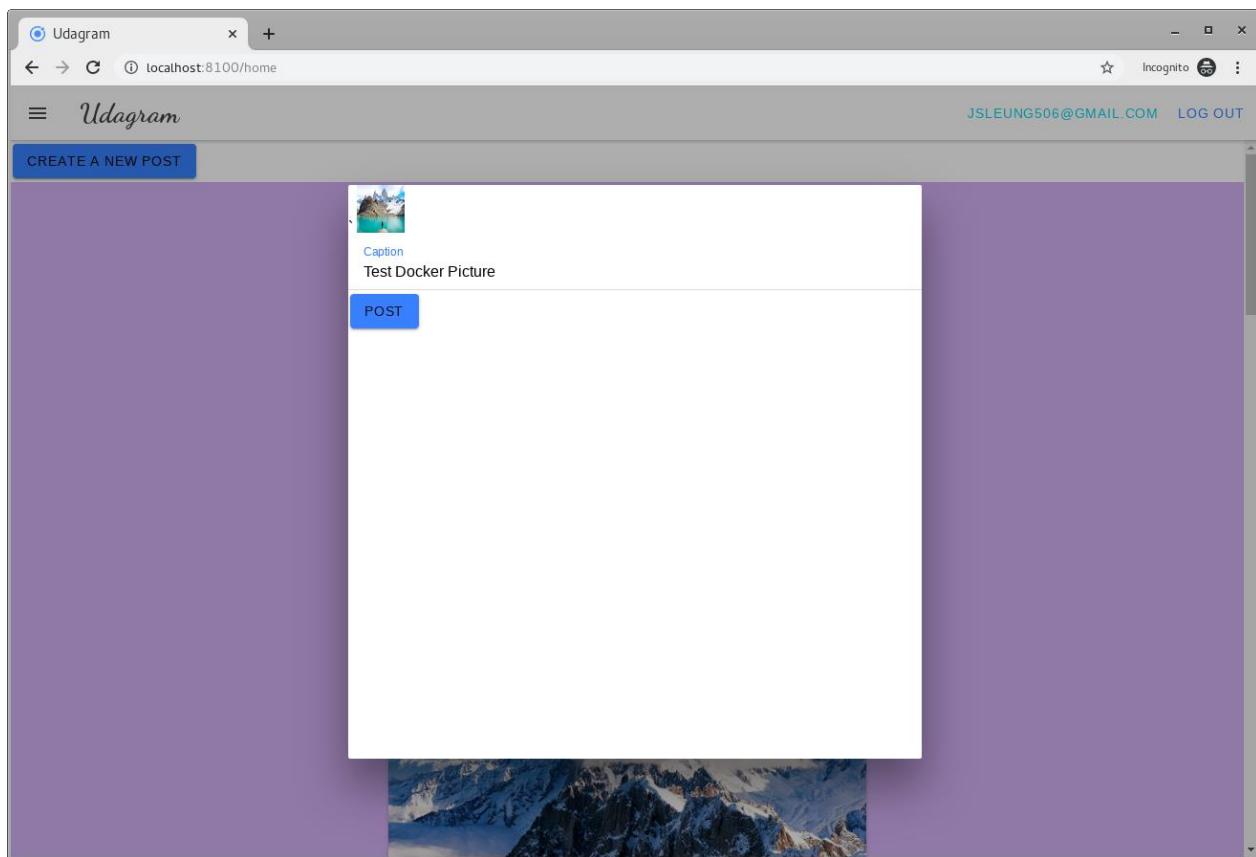
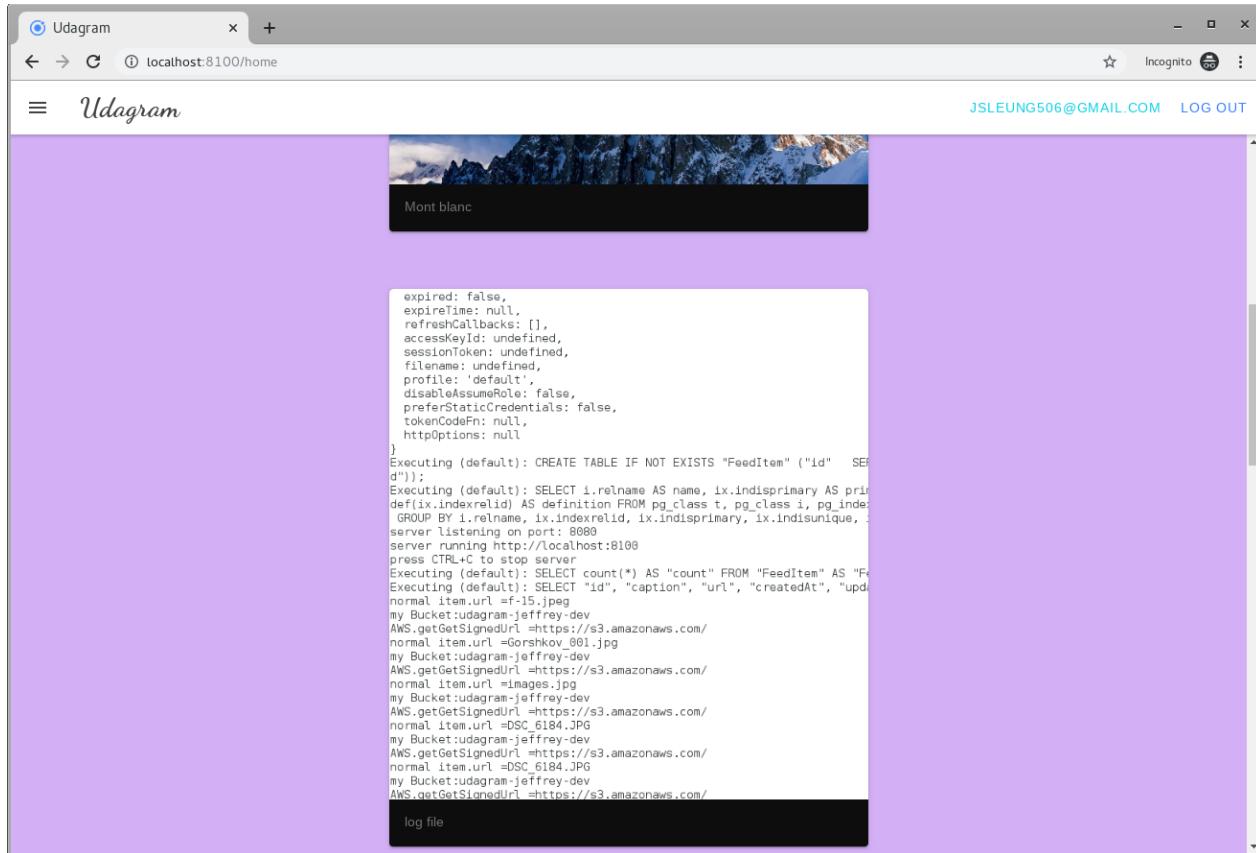
```
[vagrant@localhost docker]$ docker-compose up
/usr/lib/python2.7/site-packages/requests/_init__.py:91: RequestsDependencyWarning: urllib3 (1.25.6) or chardet (3.0.4) doesn't match a supported version!
  RequestsDependencyWarning)
Creating docker_backend-user_1 ... done
Creating docker_frontend_1 ... done
Creating docker_backend-feed_1 ... done
Creating docker_reverseproxy_1 ... done
Attaching to docker_backend-user_1, docker_backend-feed_1, docker_frontend_1, docker_reverseproxy_1
backend-user_1 | > udacity-c2-restapi@1.0.0 prod /usr/src/app
backend-user_1 | > tsc && node ./www/server.js
backend-user_1 |
backend-feed_1 | > udacity-c2-restapi@1.0.0 prod /usr/src/app
backend-feed_1 | > tsc && node ./www/server.js
backend-feed_1 |
reverseproxy_1 | 2019/11/08 14:05:14 [notice] 1#1: using the "spoll" event method
reverseproxy_1 | 2019/11/08 14:05:14 [notice] 1#1: nginx/1.17.5
reverseproxy_1 | 2019/11/08 14:05:14 [notice] 1#1: built by gcc 8.3.0 (Alpine 8.3.0)
reverseproxy_1 | 2019/11/08 14:05:14 [notice] 1#1: OS: Linux 3.10.0-514.22.2.el7.x86_64
reverseproxy_1 | 2019/11/08 14:05:14 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576/1048576
reverseproxy_1 | 2019/11/08 14:05:14 [notice] 1#1: start worker processes
reverseproxy_1 | 2019/11/08 14:05:14 [notice] 1#1: start worker process 6
backend-feed_1 | {
  dev: {
    username: 'udagramjeffrey',
    password: 'HelloWorld',
    database: 'udagram_c3_db',
    host: 'udagramdev.cj6xwvjszsvd.us-east-1.rds.amazonaws.com',
    dialect: 'postgres',
    aws_region: 'us-east-1',
    aws_profile: 'default',
    aws_media_bucket: 'udagram-jeffrey-dev',
    url: 'http://localhost:8100'
  },
  prod: {
    username: 'udagramjeffrey',
    password: 'HelloWorld',
    database: 'udagram_c3_db',
    host: 'udagramdev.cj6xwvjszsvd.us-east-1.rds.amazonaws.com',
    dialect: 'postgres',
    aws_region: 'us-east-1',
    aws_profile: 'default',
    aws_media_bucket: 'udagram-jeffrey-dev',
    url: 'http://localhost:8100'
  }
},
```

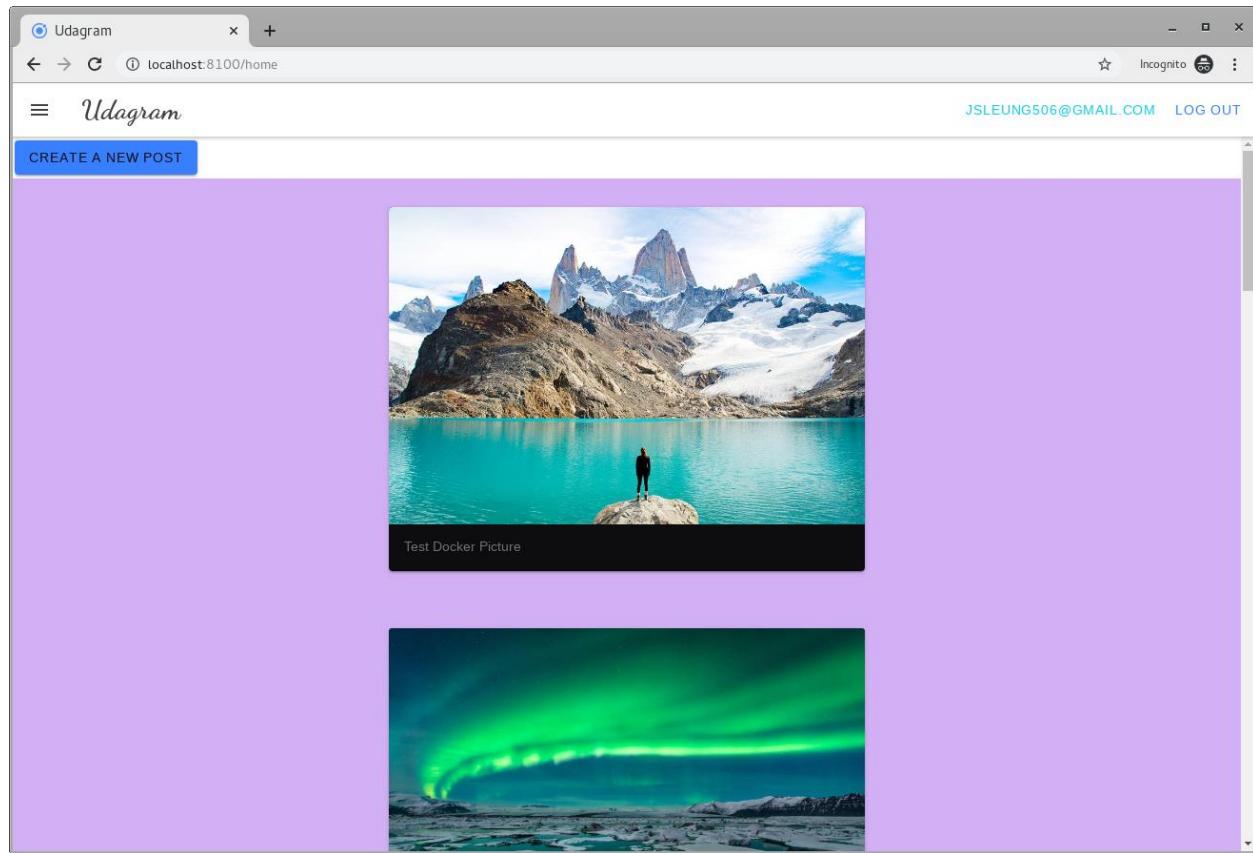


```
vagrant@localhost:~/udacity/udacity-project4/udacity-c3-deployment/docker
```

```
File Edit View Search Terminal Help
backend-user_1 |   },
backend-user_1 |   jwt: { secret: 'abcd1234' }
backend-user_1 | }
backend-feed_1 | using the following config.prod:
backend-feed_1 | {
  backend-feed_1 |   username: 'udagramjeffrey',
  backend-feed_1 |   password: 'HelloWorld',
  backend-feed_1 |   database: 'udagram_c3_db',
  backend-feed_1 |   host: 'udagramdev.cj6xwvjszsvd.us-east-1.rds.amazonaws.com',
  backend-feed_1 |   dialect: 'postgres',
  backend-feed_1 |   aws_region: 'us-east-1',
  backend-feed_1 |   aws_profile: 'default',
  backend-feed_1 |   aws_media_bucket: 'udagram-jeffrey-dev',
  backend-feed_1 |   url: 'http://localhost:8100'
  backend-feed_1 | }
  backend-feed_1 | using credentials:
  backend-feed_1 | SharedInFileCredentials {
  backend-feed_1 |   expired: false,
  backend-feed_1 |   expireTime: null,
  backend-feed_1 |   refreshCallbacks: [],
  backend-feed_1 |   accessKeyId: 'AKIA4UR3N7PEFHLFMF4P',
  backend-feed_1 |   sessionToken: undefined,
  backend-feed_1 |   filename: undefined,
  backend-feed_1 |   profile: 'default',
  backend-feed_1 |   disableAssumeRole: false,
  backend-feed_1 |   preferStaticCredentials: false,
  backend-feed_1 |   tokenCodeFn: null,
  backend-feed_1 |   httpOptions: null
  backend-feed_1 | }
  backend-user_1 | Executing (default): CREATE TABLE IF NOT EXISTS "User" ("email" VARCHAR(255), "password_hash" VARCHAR(255), "createdAt" TIMESTAMP WITH TIME ZONE, "updatedAt" TIMESTAMP WITH TIME ZONE, PRIMARY KEY ("email"));
  backend-feed_1 | Executing (default): CREATE TABLE IF NOT EXISTS "FeedItem" ("id" SERIAL, "caption" VARCHAR(255), "url" VARCHAR(255), "createdAt" TIMESTAMP WITH TIME ZONE, "updatedAt" TIMESTAMP WITH TIME ZONE, PRIMARY KEY ("id"));
  backend-user_1 | Executing (default): SELECT i.relname AS name, ix.indisprimary AS primary, ix.indisunique AS unique, ix.indkey AS indkey, array_agg(a.attname) AS column_indexes, array_agg(a.attname) AS column_names, pg_get_indexdef(ix.indexrelid) AS definition FROM pg_class t, pg_class i, pg_index ix, pg_attribute a WHERE t.oid = ix.indrelid AND i.oid = ix.indrelid AND a.attrelid = t.oid AND t.relkind = 'r' AND t.relname = 'User' GROUP BY i.relname, ix.indexrelid, ix.indisprimary, ix.indisunique, ix.indkey ORDER BY i.relname;
  backend-feed_1 | Executing (default): SELECT i.relname AS name, ix.indisprimary AS primary, ix.indisunique AS unique, ix.indkey AS indkey, array_agg(a.attname) AS column_indexes, array_agg(a.attname) AS column_names, pg_get_indexdef(ix.indexrelid) AS definition FROM pg_class t, pg_class i, pg_index ix, pg_attribute a WHERE t.oid = ix.indrelid AND i.oid = ix.indrelid AND a.attrelid = t.oid AND t.relkind = 'r' AND t.relname = 'FeedItem' GROUP BY i.relname, ix.indexrelid, ix.indisprimary, ix.indisunique, ix.indkey ORDER BY i.relname;
  backend-user_1 | restapi-user server listening on port: 8080
  backend-user_1 | server running http://localhost:8100
  backend-user_1 | press CTRL+C to stop server
  backend-feed_1 | restapi-feed server listening on port: 8080
  backend-feed_1 | server running http://localhost:8100
  backend-feed_1 | press CTRL+C to stop server
```

After the Docker images were successfully started, user can login to the Udagram application, view posts, and upload photos to create new posts:





3. Deployment

3.1 The application runs on a cluster in the cloud

Please refer to [section 2. Instructions for installing Kubernetes cluster in AWS](#) on how to create the Kubernetes cluster and [section 3. Instructions for creation of Kubernetes pods in Kubernetes cluster](#) on how to create the pods in the Kubernetes cluster.

After running Terraform, the following infrastructure are created in AWS:

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs
udacitykubone-pool1-644668b9b7-w65xv	i-06fc752cc8de94cf0	t3.medium	us-east-1a	running	2/2 checks passed	None	ec2-3-85-146-57.compute-1.amazonaws.com	3.85.146.57	-
udacitykubone-control_plane-2	i-07e2f077966b9b3f3f	t3.medium	us-east-1b	running	2/2 checks passed	None	ec2-3-88-167-211.compute-1.amazonaws.com	3.88.167.211	-
udacitykubone-pool1-644668b9b7-g4tn9	i-07ba50b4402364a93	t3.medium	us-east-1a	running	2/2 checks passed	None	ec2-3-86-110-16.compute-1.amazonaws.com	3.86.110.16	-
udacitykubone-control_plane-3	i-09c753bc4a0f7c7d7a	t3.medium	us-east-1c	running	2/2 checks passed	None	ec2-3-81-112-66.compute-1.amazonaws.com	3.81.112.66	-
udacitykubone-control_plane-1	i-09c921dd78110f5a	t3.medium	us-east-1a	running	2/2 checks passed	None	ec2-18-205-163-198.compute-1.amazonaws.com	18.205.163.198	-
udacitykubone-pool1-644668b9b7-5f8c9	i-09f64775109470f21	t3.medium	us-east-1a	running	2/2 checks passed	None	ec2-52-207-133-84.compute-1.amazonaws.com	52.207.133.84	-

After installing Kubernetes cluster in our AWS and created the pods, we execute `kubectl get pods`, which shows the pods representing the four services (backend-feed, backend-user, frontend, reverseproxy) in addition to the fluentd service for AWS CloudWatch are all running properly:

```
vagrant@localhost:~$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
backend-feed-5f96996ff5-gxqr6   1/1     Running   0          3m55s
backend-feed-5f96996ff5-hbpnq   1/1     Running   0          3m49s
backend-feed-5f96996ff5-kdqhv   1/1     Running   0          3m55s
backend-user-56dd55965-866md   1/1     Running   0          3m54s
backend-user-56dd55965-qp8c9   1/1     Running   0          3m47s
backend-user-56dd55965-wtdfh   1/1     Running   0          3m55s
fluentd-cdzgv                  1/1     Running   2          2d
fluentd-f8466                  1/1     Running   2          2d
fluentd-fgprt                  1/1     Running   1          2d
frontend-67978898c8-dmj66      1/1     Running   0          3m54s
frontend-67978898c8-wzb66      1/1     Running   0          3m54s
reverseproxy-866f574cdf-lmvf7  1/1     Running   0          3m55s
reverseproxy-866f574cdf-n7ncl  1/1     Running   0          3m55s
[vagrant@localhost ~]$
```

To verify the application behavior, we execute the following in order to use our Web Browser to access Udagram in <http://localhost:8100>:

```
kubectl port-forward service/reverseproxy 8080:8080
```

```
kubectl port-forward service/frontend 8100:8100
```

The screenshot shows a terminal window with two tabs. The left tab displays the output of `kubectl get pods` and `kubectl get svc` commands, listing various pods and services. The right tab shows the command `kubectl port-forward service/frontend 8100:8100` being run. Below the terminal is a browser window titled "Udagram" at localhost:8100/home. The browser shows a purple header bar with the Udagram logo and a "CREATE A NEW POST" button. The main content area displays two images: a large ship (the Titanic) and a mountain range (likely Mount Fitz Roy).

```
vagrant@localhost:~$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
backend-feed-5f96996ff5-gxqr6   1/1     Running   0          3m55s
backend-feed-5f96996ff5-hbpnq   1/1     Running   0          3m49s
backend-feed-5f96996ff5-kdqhv   1/1     Running   0          3m55s
backend-user-56dd55965-866md   1/1     Running   0          3m54s
backend-user-56dd55965-qpbc9   1/1     Running   0          3m47s
backend-user-56dd55965-wtdfh   1/1     Running   0          3m55s
fluentd-f8466                  1/1     Running   2          2d
fluentd-fgprt                  1/1     Running   1          2d
frontend-67978898c8-dmj66      1/1     Running   0          3m54s
frontend-67978898c8-wzb66      1/1     Running   0          3m54s
reverseproxy-866f574cdf-lmfv7  1/1     Running   0          3m47s
reverseproxy-866f574cdf-n7nc1  1/1     Running   0          3m55s
[vagrant@localhost ~]$ clear

[vagrant@localhost ~]$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
backend-feed-5f96996ff5-gxqr6   1/1     Running   0          3m55s
backend-feed-5f96996ff5-hbpnq   1/1     Running   0          3m49s
backend-feed-5f96996ff5-kdqhv   1/1     Running   0          3m55s
backend-user-56dd55965-866md   1/1     Running   0          3m54s
backend-user-56dd55965-qpbc9   1/1     Running   0          3m47s
backend-user-56dd55965-wtdfh   1/1     Running   0          3m55s
fluentd-f8466                  1/1     Running   2          2d
fluentd-fgprt                  1/1     Running   1          2d
frontend-67978898c8-dmj66      1/1     Running   0          3m54s
frontend-67978898c8-wzb66      1/1     Running   0          3m54s
reverseproxy-866f574cdf-lmfv7  1/1     Running   0          3m55s
reverseproxy-866f574cdf-n7nc1  1/1     Running   0          3m55s
[vagrant@localhost ~]$ kubectl port-forward service/reverseproxy 8080:8080
Forwarding from 127.0.0.1:8080 -> 8080
Forwarding from [::1]:8080 -> 8080
Handling connection for 8080

[vagrant@localhost ~]$
```

We can access Udagram in our Web Browser. We noted the new look of our Udagram using the new scss style that we defined in the Ionic client. The user can login, view existing posts, and also create new posts by upload new photo and enter a caption:

The screenshot shows a terminal window with two tabs. The left tab displays the output of `kubectl get pods` and `kubectl get svc` commands, listing various pods and services. The right tab shows the command `kubectl port-forward service/frontend 8100:8100` being run. Below the terminal is a browser window titled "Udagram" at localhost:8100/home. The browser shows a purple header bar with the Udagram logo and a "CREATE A NEW POST" button. The main content area displays two images: a large ship (the Titanic) and a mountain range (likely Mount Fitz Roy).

```
vagrant@localhost:~$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
backend-feed-5f96996ff5-gxqr6   1/1     Running   0          3m55s
backend-feed-5f96996ff5-hbpnq   1/1     Running   0          3m49s
backend-feed-5f96996ff5-kdqhv   1/1     Running   0          3m55s
backend-user-56dd55965-866md   1/1     Running   0          3m54s
backend-user-56dd55965-qpbc9   1/1     Running   0          3m47s
backend-user-56dd55965-wtdfh   1/1     Running   0          3m55s
fluentd-f8466                  1/1     Running   2          2d
fluentd-fgprt                  1/1     Running   1          2d
frontend-67978898c8-dmj66      1/1     Running   0          3m54s
frontend-67978898c8-wzb66      1/1     Running   0          3m54s
reverseproxy-866f574cdf-lmfv7  1/1     Running   0          3m55s
reverseproxy-866f574cdf-n7nc1  1/1     Running   0          3m55s
[vagrant@localhost ~]$ clear

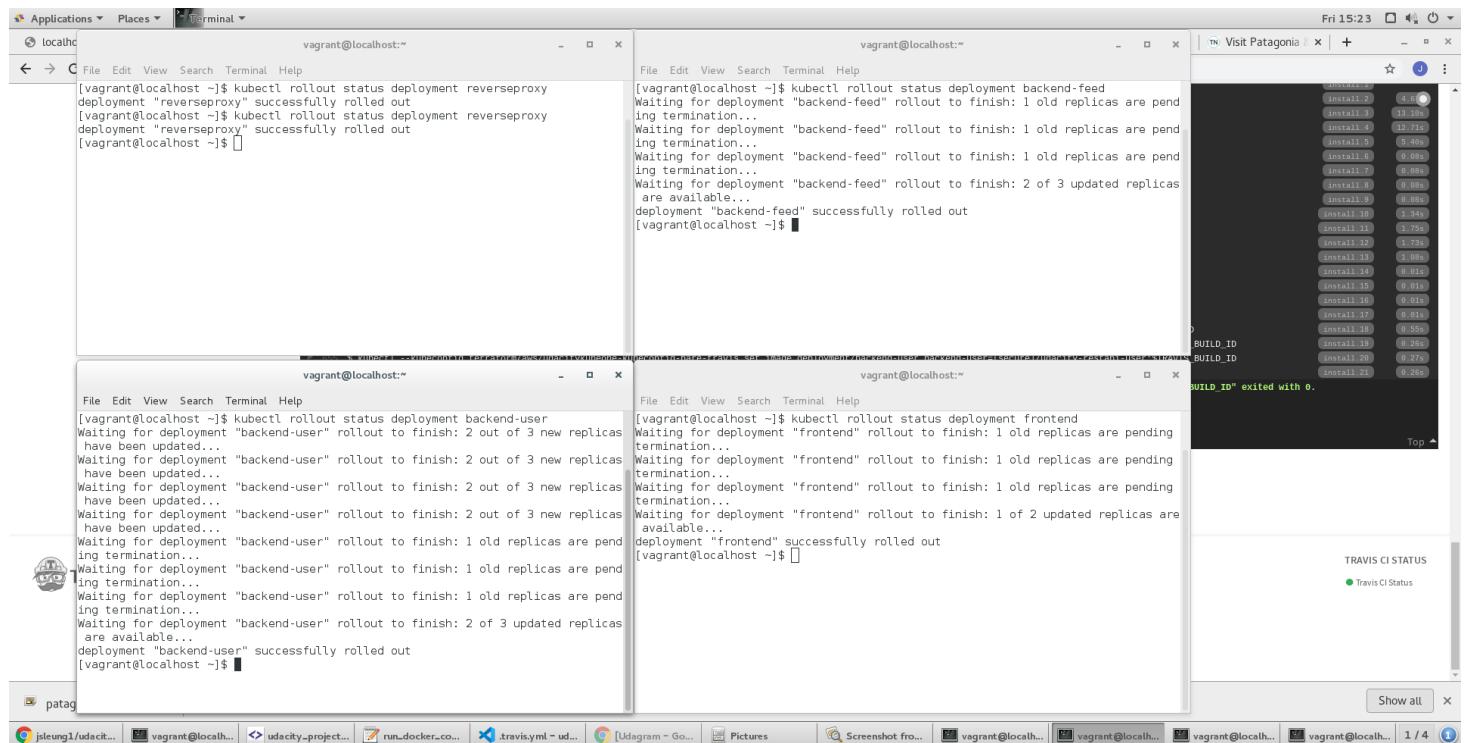
[vagrant@localhost ~]$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
backend-feed-5f96996ff5-gxqr6   1/1     Running   0          3m55s
backend-feed-5f96996ff5-hbpnq   1/1     Running   0          3m49s
backend-feed-5f96996ff5-kdqhv   1/1     Running   0          3m55s
backend-user-56dd55965-866md   1/1     Running   0          3m54s
backend-user-56dd55965-qpbc9   1/1     Running   0          3m47s
backend-user-56dd55965-wtdfh   1/1     Running   0          3m55s
fluentd-f8466                  1/1     Running   2          2d
fluentd-fgprt                  1/1     Running   1          2d
frontend-67978898c8-dmj66      1/1     Running   0          3m54s
frontend-67978898c8-wzb66      1/1     Running   0          3m54s
reverseproxy-866f574cdf-lmfv7  1/1     Running   0          3m55s
reverseproxy-866f574cdf-n7nc1  1/1     Running   0          3m55s
[vagrant@localhost ~]$ kubectl port-forward service/reverseproxy 8080:8080
Forwarding from 127.0.0.1:8080 -> 8080
Forwarding from [::1]:8080 -> 8080
Handling connection for 8080

[vagrant@localhost ~]$ kubectl port-forward service/frontend 8100:8100
Forwarding from 127.0.0.1:8100 -> 80
Forwarding from [::1]:8100 -> 80
Handling connection for 8100
```

3.2 The app can be upgraded via rolling-update

From the previous section 3.1, we have introduced a new version of the application by using rolling-update: When we deploy the new version in section 3.1, we can see Kubernetes is performing a rolling update by execute the following commands during the deployment:

```
kubectl rollout status deployment reverseproxy  
kubectl rollout status deployment backend-feed  
kubectl rollout status deployment backend-user  
kubectl rollout status deployment frontend
```



To further verify the deployments are using strategy type as RollingUpdate, we execute the following:

```
kubectl describe deployment reverseproxy  
kubectl describe deployment backend-feed  
kubectl describe deployment backend-user  
kubectl describe deployment frontend
```

which shows the deployments: frontend, reverseproxy, backend-user and backend-feed are using StrategyType as RollingUpdate:

```
vagrant@localhost:~
```

```
File Edit View Search Terminal Help
Waiting for deployment "frontend" rollout to finish: 1 of 2 updated replicas are available...
Deployment "frontend" successfully rolled out
[vagrant@localhost ~]$ kubectl describe deployment frontend
Name:           frontend
Namespace:      default
CreationTimestamp: Sun, 27 Oct 2019 14:49:39 +0000
Labels:         service=frontend
Annotations:   deployment.kubernetes.io/revision: 2
               kubectl.kubernetes.io/last-applied-configuration:
                 {"apiVersion":"extensions/v1beta1","kind":"Deployment","metadata":{"annotations":{},"labels":{"service":"frontend"},"name":"frontend","name..."}
Selector:       api=external
Replicas:       2 desired | 2 updated | 2 total | 2 available | 0 unavailable
StrategyType:  RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 1 max unavailable, 1 max surge
Pod Template:
  Labels:  api=external
          service=frontend
  Containers:
    frontend:
      Image:      jsleung1/udacity-frontend:135658293
      Port:       80/TCP
      Host Port:  0/TCP
      Limits:
        cpu:  500m
        memory: 1Gi
      Requests:
        cpu:  250m
        memory: 64Mi
      Environment: <none>
      Mounts:    <none>
      Volumes:   <none>
  Conditions:
    Type     Status  Reason
    ----     ----  -----
    Available  True   MinimumReplicasAvailable
OldReplicaSets:  <none>
NewReplicaSet:   frontend-74c94946b8 (2/2 replicas created)
Events:
  Type  Reason     Age   From            Message
  ----  ----     ----  ----
  Normal  ScalingReplicaSet  3m27s  deployment-controller  Scaled up replica set frontend-74c94946b8 to 1
  Normal  ScalingReplicaSet  3m27s  deployment-controller  Scaled down replica set frontend-84dcfc6bc4 to 1
  Normal  ScalingReplicaSet  3m27s  deployment-controller  Scaled up replica set frontend-74c94946b8 to 2
  Normal  ScalingReplicaSet  3m1s   deployment-controller  Scaled down replica set frontend-84dcfc6bc4 to 0
[vagrant@localhost ~]$
```

```
vagrant@localhost:~
```

```
File Edit View Search Terminal Help
deployment "reverseproxy" successfully rolled out
[vagrant@localhost ~]$ kubectl rollout status deployment reverseproxy
deployment "reverseproxy" successfully rolled out
[vagrant@localhost ~]$ kubectl describe deployment reverseproxy
Name:           reverseproxy
Namespace:      default
CreationTimestamp: Sun, 27 Oct 2019 14:46:17 +0000
Labels:         service=reverseproxy
Annotations:   deployment.kubernetes.io/revision: 2
               kubectl.kubernetes.io/last-applied-configuration:
                 {"apiVersion":"extensions/v1beta1","kind":"Deployment","metadata":{"annotations":{},"labels":{"service":"reverseproxy"},"name":"reverseproxy..."}
Selector:       service=reverseproxy
Replicas:       2 desired | 2 updated | 2 total | 2 available | 0 unavailable
StrategyType:  RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 1 max unavailable, 1 max surge
Pod Template:
  Labels:  service=reverseproxy
  Containers:
    reverseproxy:
      Image:      jsleung1/reverseproxy:135658293
      Port:       8080/TCP
      Host Port:  0/TCP
      Limits:
        cpu:  500m
        memory: 1Gi
      Requests:
        cpu:  250m
        memory: 64Mi
      Environment: <none>
      Mounts:    <none>
      Volumes:   <none>
  Conditions:
    Type     Status  Reason
    ----     ----  -----
    Available  True   MinimumReplicasAvailable
OldReplicaSets:  <none>
NewReplicaSet:   reverseproxy-646f8bc755 (2/2 replicas created)
Events:
  Type  Reason     Age   From            Message
  ----  ----     ----  ----
  Normal  ScalingReplicaSet  85s   deployment-controller  Scaled up replica set reverseproxy-646f8bc755 to 1
  Normal  ScalingReplicaSet  84s   deployment-controller  Scaled down replica set reverseproxy-58b8cf845c to 1
  Normal  ScalingReplicaSet  84s   deployment-controller  Scaled up replica set reverseproxy-646f8bc755 to 2
  Normal  ScalingReplicaSet  81s   deployment-controller  Scaled down replica set reverseproxy-58b8cf845c to 0
[vagrant@localhost ~]$
```

```
vagrant@localhost:~
```

```

File Edit View Search Terminal Help
Annotations: deployment.kubernetes.io/revision: 2
    kubectl.kubernetes.io/last-applied-configuration:
        {"apiVersion":"extensions/v1beta1","kind":"Deployment","metadata":{"annotations":{},"labels":{"service":"backend-user"},"name":"backend-us..."}}

Selector: service=backend-user
Replicas: 3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType: RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 1 max unavailable, 1 max surge
Pod Template:
Labels: service=backend-user
Containers:
  backend-user:
    Image: jsleungl/udacity-restapi-user:135658293
    Port: <none>
    Host Port: <none>
    Limits:
      cpu: 500m
      memory: 1Gi
    Requests:
      cpu: 250m
      memory: 64Mi
    Environment:
      URL: <set to the key 'URL' of config map 'env-config'> Optional: false
      JWT_SECRET: <set to the key 'JWT_SECRET' of config map 'env-config'> Optional: false
      POSTGRESS_DB: <set to the key 'POSTGRESS_DB' of config map 'env-config'> Optional: false
      POSTGRESS_HOST: <set to the key 'POSTGRESS_HOST' of config map 'env-config'> Optional: false
      POSTGRESS_PASSWORD: <set to the key 'POSTGRESS_PASSWORD' in secret 'env-secret'> Optional: false
      POSTGRESS_USERNAME: <set to the key 'POSTGRESS_USERNAME' in secret 'env-secret'> Optional: false
    Mounts:
    Volumes:
    Conditions:
      Type Status Reason
      ---- -----
      Available True MinimumReplicasAvailable
OldReplicaSets: <none>
NewReplicaSet: backend-user-764566d687 (3/3 replicas created)
Events:
  Type Reason Age From Message
  ---- ---- -- -- -----
  Normal ScalingReplicaSet 2m8s deployment-controller Scaled up replica set backend-user-764566d687 to 1
  Normal ScalingReplicaSet 2m8s deployment-controller Scaled down replica set backend-user-85b86f587c to 2
  Normal ScalingReplicaSet 2m8s deployment-controller Scaled up replica set backend-user-764566d687 to 2
  Normal ScalingReplicaSet 2m deployment-controller Scaled down replica set backend-user-85b86f587c to 1
  Normal ScalingReplicaSet 2m deployment-controller Scaled up replica set backend-user-764566d687 to 3
  Normal ScalingReplicaSet 116s deployment-controller Scaled down replica set backend-user-85b86f587c to 0
[vagrant@localhost ~]$ 

```

```
vagrant@localhost:~
```

```

File Edit View Search Terminal Help
[vagrant@localhost ~]$ kubectl describe deployment backend-feed
Name: backend-feed
Namespace: default
CreationTimestamp: Sun, 27 Oct 2019 14:39:17 +0000
Labels: service=backend-feed
Annotations: deployment.kubernetes.io/revision: 24
    kubectl.kubernetes.io/last-applied-configuration:
        {"apiVersion":"extensions/v1beta1","kind":"Deployment","metadata":{"annotations":{},"labels":{"service":"backend-feed"},"name":"backend-fe..."}}

Selector: service=backend-feed
Replicas: 3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType: RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 1 max unavailable, 1 max surge
Pod Template:
Labels: service=backend-feed
Containers:
  backend-feed:
    Image: jsleungl/udacity-restapi-feed:135658293
    Port: <none>
    Host Port: <none>
    Limits:
      cpu: 500m
      memory: 1Gi
    Requests:
      cpu: 250m
      memory: 64Mi
    Environment:
      URL: <set to the key 'URL' of config map 'env-config'> Optional: false
      AWS_BUCKET: <set to the key 'AWS_BUCKET' of config map 'env-config'> Optional: false
      AWS_PROFILE: <set to the key 'AWS_PROFILE' of config map 'env-config'> Optional: false
      AWS_REGION: <set to the key 'AWS_REGION' of config map 'env-config'> Optional: false
      JWT_SECRET: <set to the key 'JWT_SECRET' of config map 'env-config'> Optional: false
      POSTGRESS_DB: <set to the key 'POSTGRESS_DB' of config map 'env-config'> Optional: false
      POSTGRESS_HOST: <set to the key 'POSTGRESS_HOST' of config map 'env-config'> Optional: false
      POSTGRESS_PASSWORD: <set to the key 'POSTGRESS_PASSWORD' in secret 'env-secret'> Optional: false
      POSTGRESS_USERNAME: <set to the key 'POSTGRESS_USERNAME' in secret 'env-secret'> Optional: false
    Mounts:
    Volumes:
      aws-secret: /aws/ from aws-secret (ro)
    Conditions:
      Type Status Reason
      ---- -----
      aws-secret: Secret (a volume populated by a Secret)
      SecretName: aws-secret
      Optional: false
Conditions:
  Type Status Reason
  ---- -----

```

3.3 A/B deployment of the application

We will simulate A/B deployment of the application by perform rollback only to the frontend deployment while the reverseproxy, backend-user and backend-feed service still use the latest deployment. In section 3.1, we deployed our frontend with the new look (purple looking GUI). The frontend is using the replica set frontend-67978898c8. This can be verified by execute the following commands:

```
kubectl describe deployment frontend
```

```
kubectl get pods
```

which return the following result:

```
vagrant@localhost:~$ kubectl describe deployment frontend
  Name:           frontend
  Namespace:     default
  Labels:         app=frontend
  Annotations:   
  Selector:      app=frontend
  Replicas:      2 desired | 2 total
  Progressing:   True
  Starting:      2018-06-20T10:45:00Z
  LastTransitionTime: 2018-06-20T10:45:00Z
  Status:        Running
  Template:
    Labels:  app=frontend
    Containers:
      frontend:
        Image:      jsleung1/udacity-frontend:135660432
        Port:       80/TCP
        Host Port:  0/TCP
        Limits:
          CPU:        500m
          Memory:    1Gi
        Requests:
          CPU:        250m
          Memory:    64Mi
        Environment: <none>
        Mounts:     <none>
        Volumes:    <none>
    Conditions:
      Type  Status  Reason
      ----  -----  -----
      Available  True   MinimumReplicasAvailable
      OldReplicaSets: <none>
      NewReplicaSet:  frontend-67978898c8 (2/2 replicas created)
    Events:
      Type  Reason  Age From             Message
      ----  -----  --  --              --
      Normal  ScalingReplicaSet  28m  deployment-controller  Scaled up replica set frontend-74c94946b8 to 1
      Normal  ScalingReplicaSet  28m  deployment-controller  Scaled down replica set frontend-84dcfcf6b4 to 1
      Normal  ScalingReplicaSet  28m  deployment-controller  Scaled up replica set frontend-74c94946b8 to 2
      Normal  ScalingReplicaSet  28m  deployment-controller  Scaled down replica set frontend-84dcfcf6b4 to 0
      Normal  ScalingReplicaSet  16m  deployment-controller  Scaled up replica set frontend-67978898c8 to 1
      Normal  ScalingReplicaSet  16m  deployment-controller  Scaled down replica set frontend-74c94946b8 to 1
      Normal  ScalingReplicaSet  16m  deployment-controller  Scaled up replica set frontend-67978898c8 to 2
      Normal  ScalingReplicaSet  15m  deployment-controller  Scaled down replica set frontend-74c94946b8 to 0
  vagrant@localhost ~ $ kubectl get pods
  NAME                    READY  STATUS    RESTARTS  AGE
  backend-feed-5f96996ff5-gxqr6  1/1   Running   0          16m
  backend-feed-5f96996ff5-hbpnq  1/1   Running   0          16m
  backend-feed-5f96996ff5-kdqhv  1/1   Running   0          16m
  backend-user-56dd55965-866md  1/1   Running   0          16m
  backend-user-56dd55965-qp8c9  1/1   Running   0          16m
  backend-user-56dd55965-wtffn  1/1   Running   0          16m
  fluentd-cd2gv               1/1   Running   2          2d1h
  fluentd-f8466               1/1   Running   2          2d1h
  fluentd-fgprt               1/1   Running   1          2d1h
  Frontend-67978898c8-dmj66   1/1   Running   0          16m
  Frontend-67978898c8-wzb66   1/1   Running   0          16m
  reverseproxy-8661574cdf-lmfv7 1/1   Running   0          16m
  reverseproxy-8661574cdf-n7n61 1/1   Running   0          16m
  vagrant@localhost ~ $ kubectl rollout undo deployment frontend
```

To perform rollback of frontend deployment, we execute:

```
kubectl rollout undo deployment frontend
```

which will rollback the deployment of the frontend in section 3.1.

To confirm the rollback of the frontend deployment is completed, we execute the commands:

```
kubectl describe deployment frontend
```

```
kubectl get pods
```

which confirmed the frontend deployment is rollback to the previous replica set frontend-74c94946b8. The rollback results are shown in the next page:

```
vagrant@localhost:~
```

```

File Edit View Search Terminal Help
Labels:           service=frontend
Annotations:     deployment.kubernetes.io/revision: 4
                  kubectl.kubernetes.io/last-applied-configuration:
                  {"apiVersion":"extensions/v1beta1","kind":"Deployment","metadata":{"annotations":{},"labels":{"service":"frontend","name":"frontend","nam...}}
Selector:        apiExternal,service=frontend
Replicas:        2 desired | 2 updated | 2 total | 2 available | 0 unavailable
StrategyType:    RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 1 max unavailable, 1 max surge
Pod Template:
  Labels:  api=external
            service=frontend
  Containers:
    frontend:
      Image:  jsteungl/udacity-frontend:135658293
      Port:  80/TCP
      Host Port:  0/TCP
      Limits:
        cpu:  500m
        memory:  1Gi
      Requests:
        cpu:  250m
        memory:  64Mi
      Environment:  <none>
      Mounts:  <none>
      Volumes:  <none>
  Conditions:
    Type Status Reason
    ---- ---- -
    Available True  MinimumReplicasAvailable
OldReplicaSets:  <none>
NewReplicaSet:  frontend-74c94946b8 (2/2 replicas created)
Events:
  Type  Reason          Age   From            Message
  ----  ----          --  --   --           --
  Normal ScalingReplicaSet 32m  deployment-controller  Scaled down replica set frontend-84dcfc6bc4 to 1
  Normal ScalingReplicaSet 31m  deployment-controller  Scaled down replica set frontend-84dcfc6bc4 to 0
  Normal ScalingReplicaSet 19m  deployment-controller  Scaled up replica set frontend-67978898c8 to 2
  Normal ScalingReplicaSet 19m  deployment-controller  Scaled up replica set frontend-67978898c8 to 1
  Normal ScalingReplicaSet 19m  deployment-controller  Scaled down replica set frontend-74c94946b8 to 1
  Normal ScalingReplicaSet 19m  deployment-controller  Scaled down replica set frontend-74c94946b8 to 0
  Normal ScalingReplicaSet 113s (x2 over 32m) deployment-controller  Scaled up replica set frontend-74c94946b8 to 2
  Normal ScalingReplicaSet 113s (x2 over 32m) deployment-controller  Scaled up replica set frontend-74c94946b8 to 1
  Normal ScalingReplicaSet 113s  deployment-controller  Scaled down replica set frontend-67978898c8 to 1
  Normal ScalingReplicaSet 113s  deployment-controller  Scaled down replica set frontend-67978898c8 to 0
[vagrant@localhost ~]$ 
```

```
vagrant@localhost:~
```

```

File Edit View Search Terminal Help
OldReplicaSets:  <none>
NewReplicaSet:  frontend-67978898c8 (2/2 replicas created)
Events:
  Type  Reason          Age   From            Message
  ----  ----          --  --   --
  Normal ScalingReplicaSet 28m  deployment-controller  Scaled up replica set frontend-74c94946b8 to 1
  Normal ScalingReplicaSet 28m  deployment-controller  Scaled down replica set frontend-84dcfc6bc4 to 1
  Normal ScalingReplicaSet 28m  deployment-controller  Scaled up replica set frontend-74c94946b8 to 2
  Normal ScalingReplicaSet 28m  deployment-controller  Scaled down replica set frontend-84dcfc6bc4 to 0
  Normal ScalingReplicaSet 16m  deployment-controller  Scaled up replica set frontend-67978898c8 to 1
  Normal ScalingReplicaSet 16m  deployment-controller  Scaled down replica set frontend-74c94946b8 to 1
  Normal ScalingReplicaSet 16m  deployment-controller  Scaled up replica set frontend-67978898c8 to 2
  Normal ScalingReplicaSet 15m  deployment-controller  Scaled down replica set frontend-74c94946b8 to 0
[vagrant@localhost ~]$ kubectl pods
NAME                      READY   STATUS    RESTARTS   AGE
backend-feed-5f96996ff5-gxqr6  1/1    Running   0          16m
backend-feed-5f96996ff5-hbpnq  1/1    Running   0          16m
backend-feed-5f96996ff5-kdqhv  1/1    Running   0          16m
backend-user-56dd55965-866md  1/1    Running   0          16m
backend-user-56dd55965-qp8c9  1/1    Running   0          16m
backend-user-56dd55965-wtdfh  1/1    Running   0          16m
fluentd-cdgv                 1/1    Running   2          2d1h
fluentd-f8466                 1/1    Running   2          2d1h
fluentd-fgprt                 1/1    Running   1          2d1h
frontend-67978898c8-dmj66    1/1    Running   0          16m
frontend-67978898c8-wzb66    1/1    Running   0          16m
reverseproxy-866f574cdf-lmfv7 1/1    Running   0          16m
reverseproxy-866f574cdf-n7ncl 1/1    Running   0          16m
[vagrant@localhost ~]$ kubectl rollout undo deployment frontend
deployment.extensions/frontend rolled back
[vagrant@localhost ~]$ kubectl get pods
NAME                      READY   STATUS    RESTARTS   AGE
backend-feed-5f96996ff5-gxqr6  1/1    Running   0          17m
backend-feed-5f96996ff5-hbpnq  1/1    Running   0          17m
backend-feed-5f96996ff5-kdqhv  1/1    Running   0          17m
backend-user-56dd55965-866md  1/1    Running   0          17m
backend-user-56dd55965-qp8c9  1/1    Running   0          17m
backend-user-56dd55965-wtdfh  1/1    Running   0          17m
fluentd-cdgv                 1/1    Running   2          2d1h
fluentd-f8466                 1/1    Running   2          2d1h
fluentd-fgprt                 1/1    Running   1          2d1h
Frontend-74c94946b8-qtncc5  1/1    Running   0          5s
Frontend-74c94946b8-txtmx    1/1    Running   0          5s
reverseproxy-866f574cdf-lmfv7 1/1    Running   0          17m
reverseproxy-866f574cdf-n7ncl 1/1    Running   0          17m
[vagrant@localhost ~]$ 
```

We use our Web Browser to access Udagram using <http://localhost:8100>, which confirmed the Ionic client is restored to the original version without our new scss style:

Screenshot of a terminal window showing Kubernetes logs and a browser window displaying the Udagram application.

```
vagrant@localhost:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
backend-feed-5f96996ff5-gxqr6      1/1    Running   0          3m55s
backend-feed-5f96996ff5-hbpnq      1/1    Running   0          3m49s
backend-feed-5f96996ff5-kdqhv      1/1    Running   0          3m55s
backend-user-56dd55965-866md       1/1    Running   0          3m54s
backend-user-56dd55965-qpc89       1/1    Running   0          3m47s
backend-user-56dd55965-wtdfh       1/1    Running   0          3m55s
fluentd-cdcgj                    1/1    Running   2          2d
fluentd-784f6                   1/1    Running   2          2d
fluentd-dcrt                      1/1    Running   1          2d
frontend-67978898c8-dmj66        1/1    Running   0          3m54s
frontend-67978898c8-wzb66        1/1    Running   0          3m54s
reverseproxy-866f574cdf-1mfv7     1/1    Running   0          3m55s
reverseproxy-866f574cdf-n7ncl      1/1    Running   0          3m55s
[vagrant@localhost ~]$ clear
```

```
vagrant@localhost:~$ kubectl port-forward service/reverseproxy 8080:8080
Forwarding from 127.0.0.1:8080 -> 8080
Forwarding from [::1]:8080 -> 8080
Handling connection for 8080
Handling connection for 8080
```

Udagram - localhost:8100/home

CREATE A NEW POST

Titanic

1 / 4

3.4 The application is monitored by Amazon CloudWatch

Please refer to [section 5. Instructions for installing FluentD to redirect Kubernetes logs to AWS Cloudwatch](#) on how to install FluentD to redirect Kubernetes log to AWS CloudWatch.

After getting FluentD running in the Kubernetes cluster, it will redirect the logs of our Kubernetes pods to AWS Cloudwatch under Log Group “udacity_project4_log_group”:

Screenshot of the AWS CloudWatch Management console showing the Log Groups page.

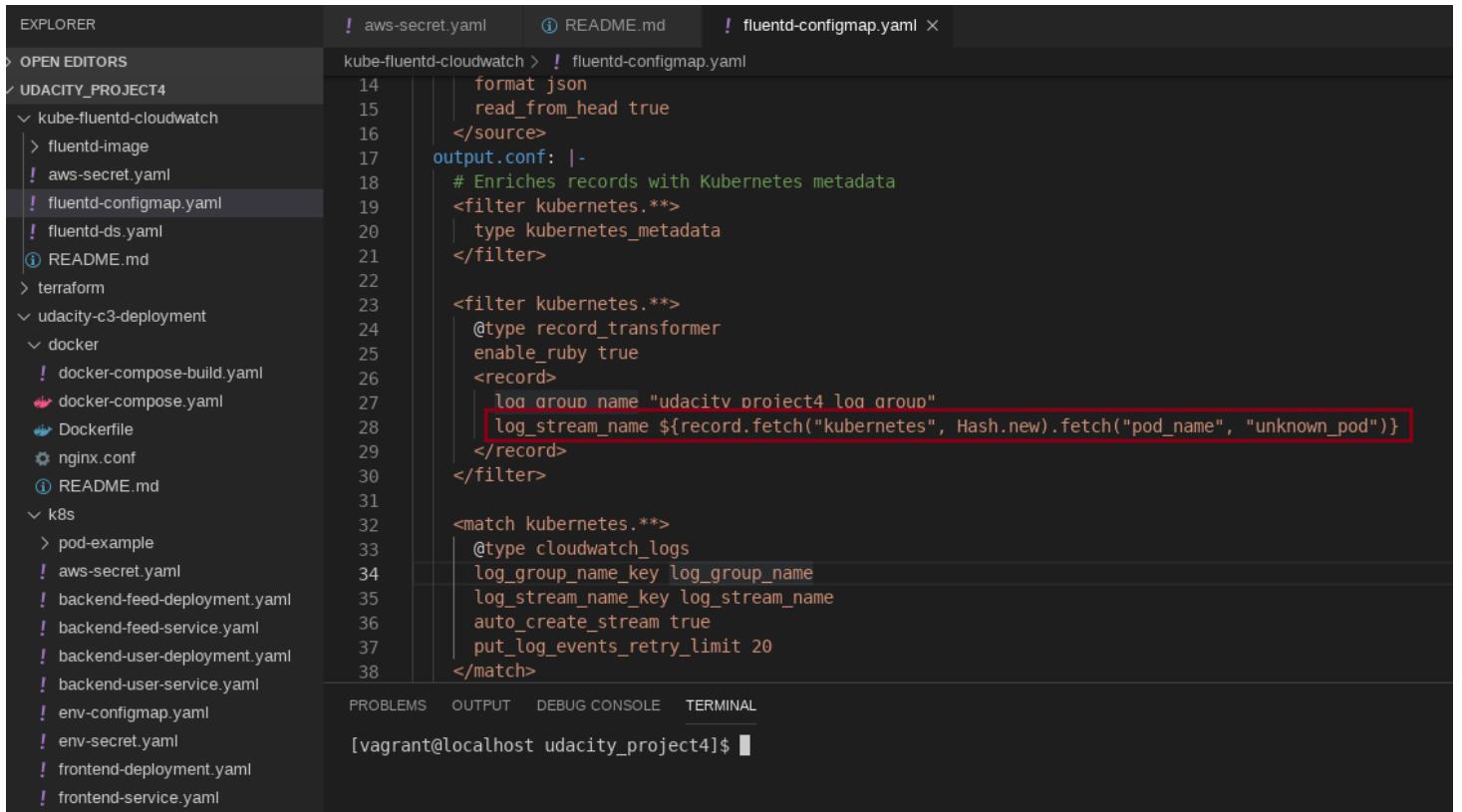
CloudWatch > Log Groups > Streams for udacity_project4_log_group

Search Log Group Create Log Stream Delete Log Stream

Filter: Log Stream Name Prefix

	Last Event Time
canal-tdkrw	2019-11-08 15:54 UTC
frontend-74c94946b8-qthnc	2019-11-08 15:54 UTC
canal-8gzkf	2019-11-08 15:50 UTC
backend-feed-5f96996ff5-kdqhv	2019-11-08 15:46 UTC
frontend-67978898c8-dmj66	2019-11-08 15:46 UTC
backend-feed-5f96996ff5-hbpnq	2019-11-08 15:37 UTC
backend-user-56dd55965-qpc89	2019-11-08 15:37 UTC
backend-user-56dd55965-866md	2019-11-08 15:37 UTC
backend-feed-5f96996ff5-gxqr6	2019-11-08 15:36 UTC
backend-user-56dd55965-wtdfh	2019-11-08 15:36 UTC
reverseproxy-646f8bc755-glmxz	2019-11-08 15:36 UTC
reverseproxy-866f574cdf-n7ncl	2019-11-08 15:36 UTC
reverseproxy-866f574cdf-lmvf7	2019-11-08 15:36 UTC
reverseproxy-646f8bc755-wccq2	2019-11-08 15:36 UTC
backend-user-764566d687-8jrcz	2019-11-08 15:24 UTC
backend-feed-7456454b8f-gld9r	2019-11-08 15:24 UTC
backend-feed-7456454b8f-jpdw	2019-11-08 15:24 UTC
backend-user-764566d687-sz62h	2019-11-08 15:24 UTC
backend-feed-7456454b8f-cmcnq	2019-11-08 15:24 UTC
backend-user-764566d687-4kwq8	2019-11-08 15:23 UTC
metrics-server-57b668cd67-st7hh	2019-11-08 15:23 UTC
reverseproxy-5bb8cf845c-wknrd	2019-11-08 15:23 UTC

Note that I have configured FluentD (using `kube-fluentd-cloudwatch/fluentd-config-map.yaml`) to use the pod name as the name of the log stream:

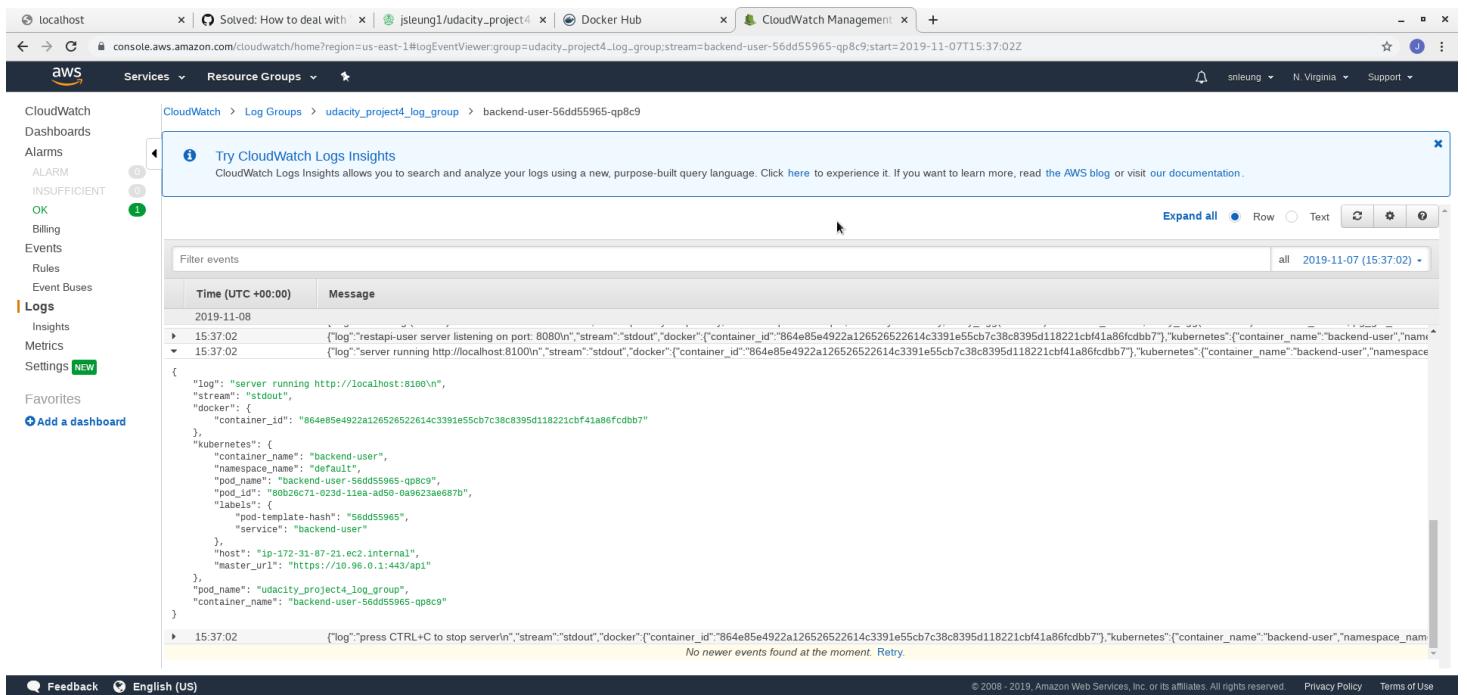


```
format json
read_from_head true
</source>
output.conf: |-
  # Enriches records with Kubernetes metadata
  <filter kubernetes.**>
    type kubernetes_metadata
  </filter>

  <filter kubernetes.**>
    @type record_transformer
    enable_ruby true
    <record>
      log_group_name "udacity_project4_log_group"
      log_stream_name ${record.fetch("kubernetes", Hash.new).fetch("pod_name", "unknown_pod")}
    </record>
  </filter>

  <match kubernetes.**>
    @type cloudwatch_logs
    log_group_name_key log_group_name
    log_stream_name_key log_stream_name
    auto_create_stream true
    put_log_events_retry_limit 20
  </match>
```

One can inspect the log message of the Kubernetes pod in AWS CloudWatch:



The screenshot shows the AWS CloudWatch Management console. On the left, there's a sidebar with navigation links like Services, Resource Groups, CloudWatch, Dashboards, Alarms, ALARM INSUFFICIENT, Billing, Events, Rules, Event Buses, Logs (which is selected), Insights, Metrics, Settings, Favorites, and Add a dashboard. The main area shows the CloudWatch Logs Insights interface. At the top, it says "Try CloudWatch Logs Insights" and provides a link to the documentation. Below that is a "Filter events" section with a "Time (UTC +00:00)" dropdown set to "2019-11-08" and a "Message" dropdown set to "all 2019-11-07 (15:37.02)". A table lists log events from November 7, 2019, at 15:37:02 UTC. The first event is expanded, showing detailed log entries. The expanded log entry includes fields like log, stream, docker, container_id, kubernetes, and host. The log message itself is a JSON object containing details about the server starting up and listening on port 8080.

To create alert notification for the end user, we create a metric filter for the log group "udacity_project4_log_group":

The screenshot shows the AWS CloudWatch Metrics Filter creation interface. On the left sidebar, under the 'Logs' section, there is a 'Create Metric Filter' button. The main area displays a table of log groups. One row is selected, showing details like Insights, Expire Events After, Metric Filters, and Subscriptions. The 'udacity_project4_log_group' is highlighted.

The screenshot shows the 'Define Logs Metric Filter' wizard. Step 1: Define Pattern shows a 'Filter Pattern' input field with 'log' typed in. Step 2: Assign Metric shows a 'Select Log Data to Test' section with a dropdown set to 'backend-feed-5f96996ff5-gxqr6'. A large text area displays sample log data. Below it, the 'Results' section states 'Found 50 matches out of 50 event(s) in the sample log.' At the bottom right are 'Cancel' and 'Assign Metric' buttons.

The screenshot shows the AWS CloudWatch Metrics Filter creation interface. On the left sidebar under 'Logs', 'Metrics' is selected. In the main area, 'Log Groups' is selected, showing 'Filters for udacity_project4_log_group'. A modal window titled 'Add Metric Filter' is open, displaying a success message: 'Your filter log has been created.' Below this, the filter details are listed: 'Filter Name: log', 'Filter Pattern: log', 'Metric: LogMetrics / log', 'Metric Value: 1', and 'Default Value: none'. There are 'Create Alarm' and edit/cancel buttons at the top right of the modal.

In the next page, we will create the alarm to notify users subscribed to the AWS Simple Notification Service (SNS) if the log count of our Kubernetes cluster exceeded 100 within 5 minutes:

The screenshot shows the 'Configure actions' step of creating a new alarm. The left sidebar lists steps: Step 1 (Specify metric and conditions), Step 2 (Configure actions - currently selected), Step 3 (Add a description), and Step 4 (Preview and create). The main area is titled 'Configure actions' and contains a 'Notification' section. It asks 'Whenever this alarm state is...' and provides three options: 'in Alarm' (selected), 'OK', and 'INSUFFICIENT_DATA'. Below this, it says 'Select an SNS topic' and shows a dropdown menu with 'Default_CloudWatch_Alarms_Topic' selected. A note below the dropdown says 'Only email lists for this account are available'. An 'Email (endpoints)' field contains 'jsleung1@yahoo.com' with a link to 'View in SNS Console'. At the bottom is a large 'Add notification' button.

localhost | Solved: How | jsleung1/ud | Docker Hub | CloudWatch | CloudWatch | Simple Noti | + | - | □ | X

console.aws.amazon.com/cloudwatch/home?region=us-east-1#alarmsV2:create/\$7B\$22metrics\$22\$3A\$5B\$5B\$22LogMetrics\$22\$2C\$22log\$22... ☆ J :

AWS Services Resource Groups

Graph
This alarm will trigger when the blue line goes above the red line for 1 datapoints within 5 minutes

Step 3
Add a description

Step 4
Preview and create

No unit
1.00k
800
600
400
200
14:00 15:00 16:00
log

Namespace
LogMetrics

Metric name
log

Statistic
Sum

Period
5 minutes

Conditions

Threshold type
 Static Use a value as a threshold Anomaly detection Use a band as a threshold

Whenever log is... Define the alarm condition
 Greater > threshold Greater/Equal >= threshold Lower/Equal <= threshold Lower < threshold

than... Define the threshold value
 Must be a number

Feedback English (US) © 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

The Alarm is triggered when the log count exceeded 100 in our Kubernetes cluster within 5 minutes:

localhost | Solved: How | jsleung1/ud | Docker Hub | CloudWatch | CloudWatch | Simple Noti | + | - | □ | X

console.aws.amazon.com/cloudwatch/home?region=us-east-1#alarmsV2:alarm/Udacity+Project+4+Kubernetes+Log+Alarm ☆ J :

AWS Services Resource Groups

CloudWatch Dashboards Alarms ALARM 1 INSUFFICIENT 0 OK 1 Billing Events Rules Event Buses Logs Insights Metrics Settings NEW Favorites Add a dashboard

CloudWatch > Alarms > Udacity Project 4 Kubernetes Log Alarm

Udacity Project 4 Kubernetes Log Alarm Edit Delete Copy

Graph View in metrics

log
log > 100 for 1 datapoints within 5 minutes

No unit
1.00k
551
100
13:30 13:45 14:00 14:15 14:30 14:45 15:00 15:15 15:30 15:45 16:00 16:15
log

In alarm

Details

Name	Threshold	Namespace	Datapoints to alarm
Udacity Project 4 Kubernetes Log Alarm	log > 100 for 1 datapoints within 5 minutes	LogMetrics	1 out of 1
Description	Last change	Metric name	Missing data treatment
No description	2019-11-08 16:28:41	log	Treat missing data as missing
Statistic	Sum	Percentiles	Percentiles with low samples

Feedback English (US) © 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

We received the alarm from AWS CloudWatch in the email:

The screenshot shows a Yahoo Mail inbox with 999+ unread messages. The main pane displays an email from AWS Notifications regarding a CloudWatch alarm. The email subject is "ALARM: \"Udacity Project 4 Kubernetes Log Alarm\" in US East (N. Virginia)". The message body provides details about the alarm, including its name, state change, reason, timestamp, and AWS account. It also includes a link to the AWS Management Console for further investigation.

Inbox 999+

Compose

Jeffrey Home

Compose

Inbox 999+

Unread Starred Drafts Sent Archive Spam Trash

Less Views Hide

Photos Documents Deals Receipts Groceries Travel

Folders Hide

+ New Folder 2019 Run... Bill payment 1 BOCHK... Coral Ree... Donations 2 Update time zone

https://mail.yahoo.com/d/folders/1/messages/APVp-iEzSTOoXcWXugyeeD12rDw?.intl=zh&.lang=zh-Hant-HK

Nespresso Jeffrey, 網上獨家優惠 只限4天 限量版Essenza Mini... 立即購買 Ad

AWS Notifications ALARM: "Udacity Project 4 Kubernetes Log Alarm" in US East (N.... 12:28 AM

AWS Notifications AWS Notification - Subscription Confirmatio... You have chosen to ... 12:08 AM

Thursday

Nikita @ LinkedIn Lea... Jeffrey. redeem this unlocked course - offer expires in 7 da... Jeffr... Nov 7

ALARM: "Udacity Project 4 Kubernetes Log Alarm" in US East (N. Virginia) Yahoo/Inbox ★

AWS Notifications <no-reply@sns.amazonaws.com> Nov 9 at 12:28 AM ★
To: jsleung1@yahoo.com

You are receiving this email because your Amazon CloudWatch Alarm "Udacity Project 4 Kubernetes Log Alarm" in the US East (N. Virginia) region has entered the ALARM state, because "Threshold Crossed: 1 out of the last 1 datapoints [951.0 (08/11/19 16:23:00)] was greater than the threshold (100.0) (minimum 1 datapoint for OK -> ALARM transition)." at "Friday 08 November, 2019 16:28:41 UTC".

View this alarm in the AWS Management Console:
<https://us-east-1.console.aws.amazon.com/cloudwatch/home?region=us-east-1#Alarms&alarm=Udacity%20Project%204%20Kubernetes%20Log%20Alarm>

Alarm Details:

- Name: Udacity Project 4 Kubernetes Log Alarm
- Description:
- State Change: INSUFFICIENT_DATA -> ALARM
- Reason for State Change: Threshold Crossed: 1 out of the last 1 datapoints [951.0 (08/11/19 16:23:00)] was greater than the threshold (100.0) (minimum 1 datapoint for OK -> ALARM transition).
- Timestamp: Friday 08 November, 2019 16:28:41 UTC
- AWS Account: 868781784008

Threshold:

- The alarm is in the ALARM state when the metric is GreaterThanThreshold 100.0 for 300 seconds.

Monitored Metric:

- MetricNamespace: LogMetrics
- MetricName: log
- Dimensions:
- Period: 300 seconds
- Statistic: Sum