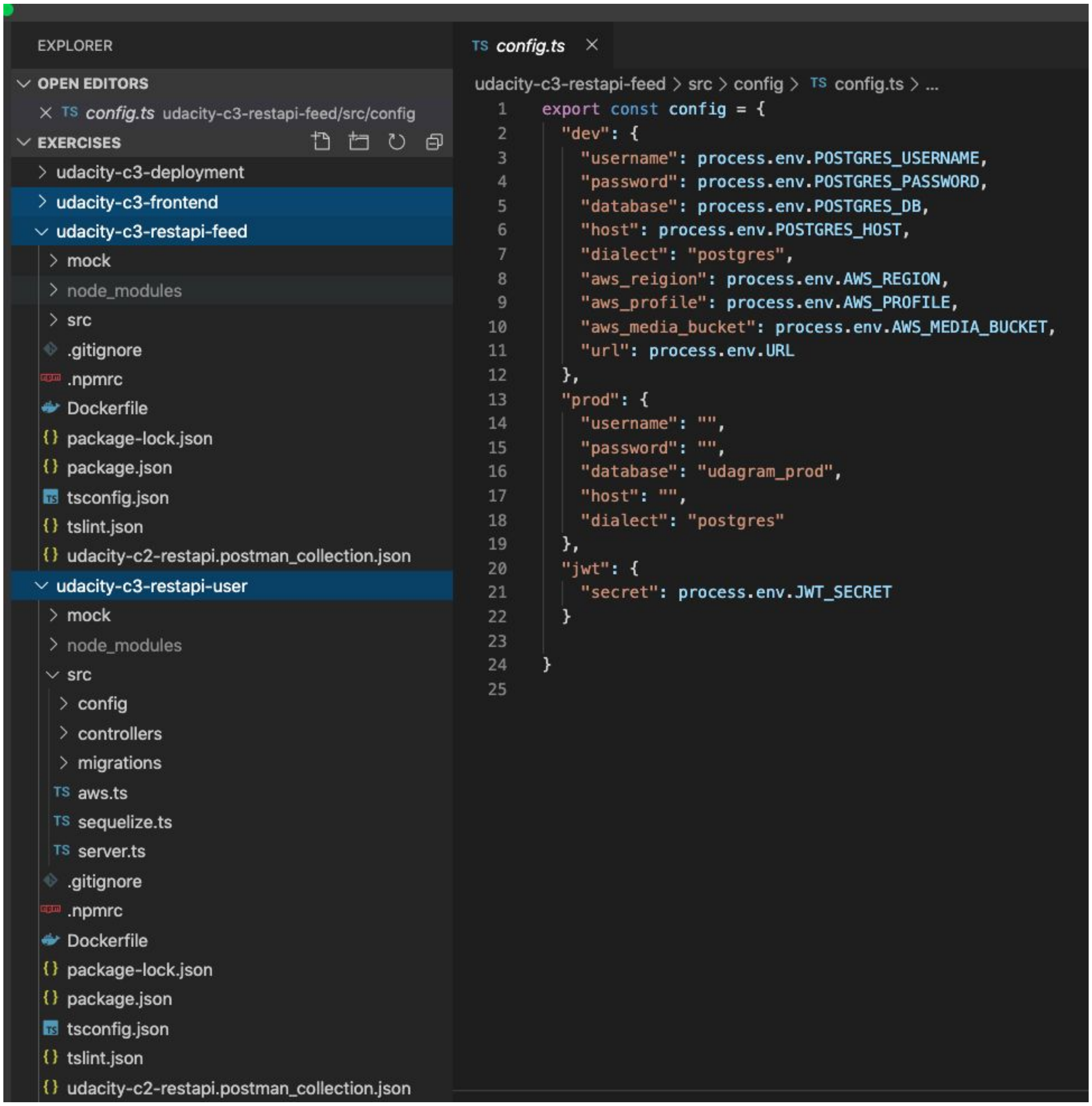
Name: Jaime Valencia
Program: CLOUD DEVELOPER
Project: #3 **Refactor Udagram Project**

1. Refactor the API



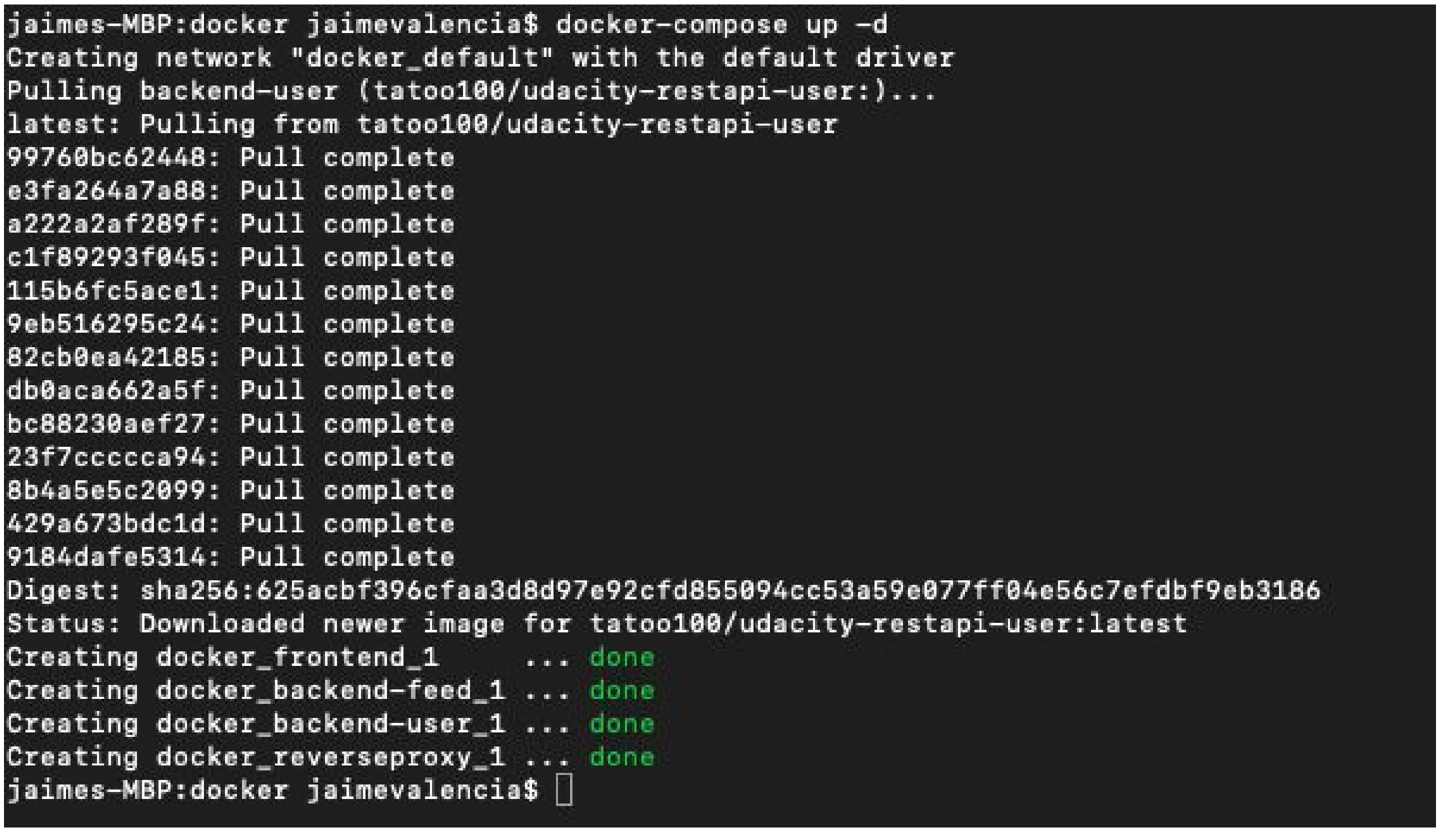
```
export const config = {
  "dev": {
    "username": process.env.POSTGRES_USERNAME,
    "password": process.env.POSTGRES_PASSWORD,
    "database": process.env.POSTGRES_DB,
    "host": process.env.POSTGRES_HOST,
    "dialect": "postgres",
    "aws_reigion": process.env.AWS_REGION,
    "aws_profile": process.env.AWS_PROFILE,
    "aws_media_bucket": process.env.AWS_MEDIA_BUCKET,
    "url": process.env.URL
  },
  "prod": {
    "username": "",
    "password": "",
    "database": "udagram_prod",
    "host": "",
    "dialect": "postgres"
  },
  "jwt": {
    "secret": process.env.JWT_SECRET
  }

}
```

2. Containerize the Code

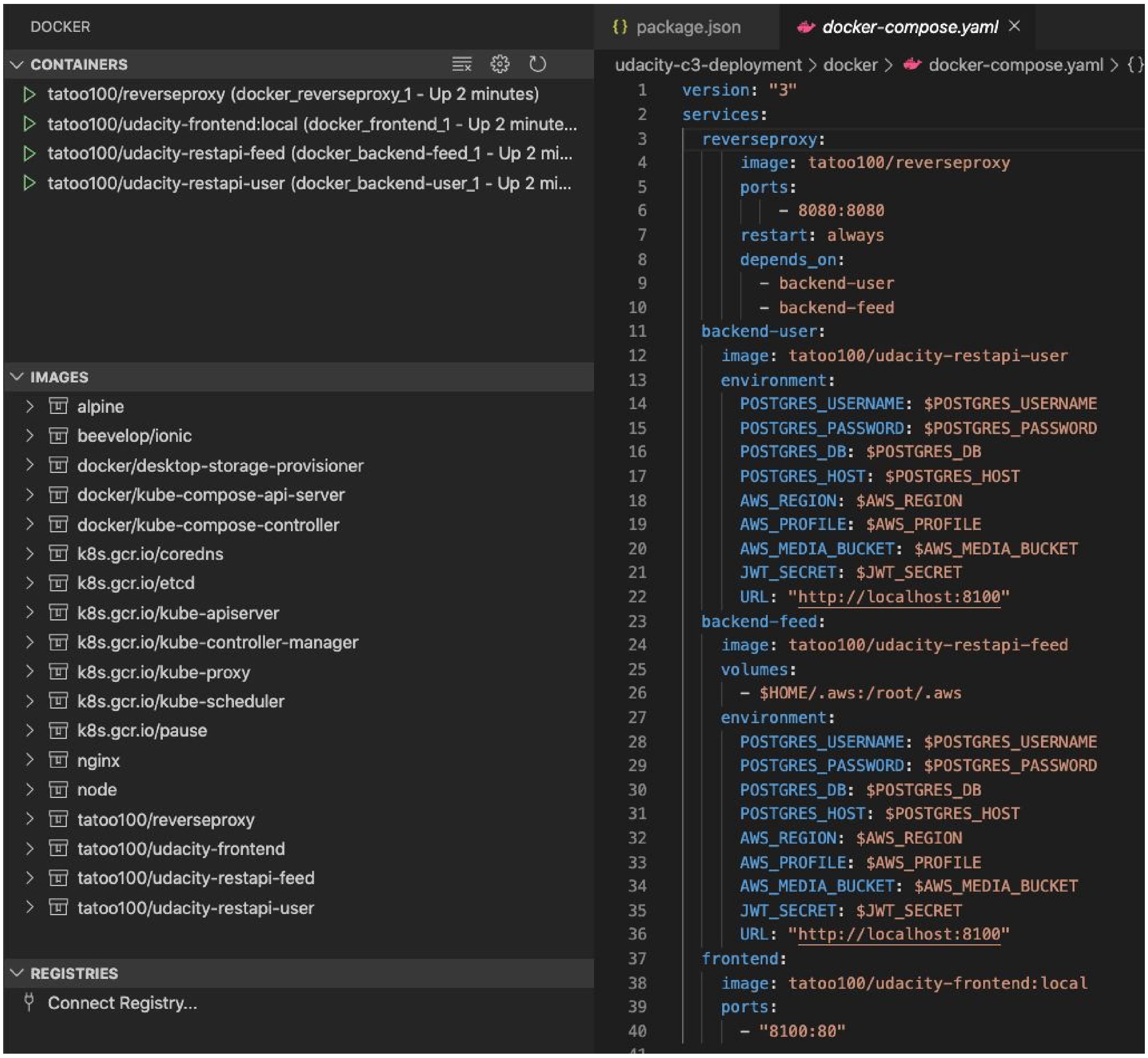
I had an issue with the backend-user image with the bcrypt library, so I had to change some of the libraries versions and build a new image and push to DockerHub.

Run command: `docker-compose up -d`



```
jaimes-MBP:docker jaimevalencia$ docker-compose up -d
Creating network "docker_default" with the default driver
Pulling backend-user (tatoo100/udacity-restapi-user:)...
latest: Pulling from tatoo100/udacity-restapi-user
99760bc62448: Pull complete
e3fa264a7a88: Pull complete
a222a2af289f: Pull complete
c1f89293f045: Pull complete
115b6fc5ace1: Pull complete
9eb516295c24: Pull complete
82cb0ea42185: Pull complete
db0aca662a5f: Pull complete
bc88230aef27: Pull complete
23f7ccccca94: Pull complete
8b4a5e5c2099: Pull complete
429a673bdc1d: Pull complete
9184dafe5314: Pull complete
Digest: sha256:625acbf396cfaa3d8d97e92cfd855094cc53a59e077ff04e56c7efdbf9eb3186
Status: Downloaded newer image for tatoo100/udacity-restapi-user:latest
Creating docker_frontend_1     ... done
Creating docker_backend-feed_1 ... done
Creating docker_backend-user_1 ... done
Creating docker_reverseproxy_1 ... done
jaimes-MBP:docker jaimevalencia$ 
```

See next picture showing all docker containers running, after running docker-compose command.

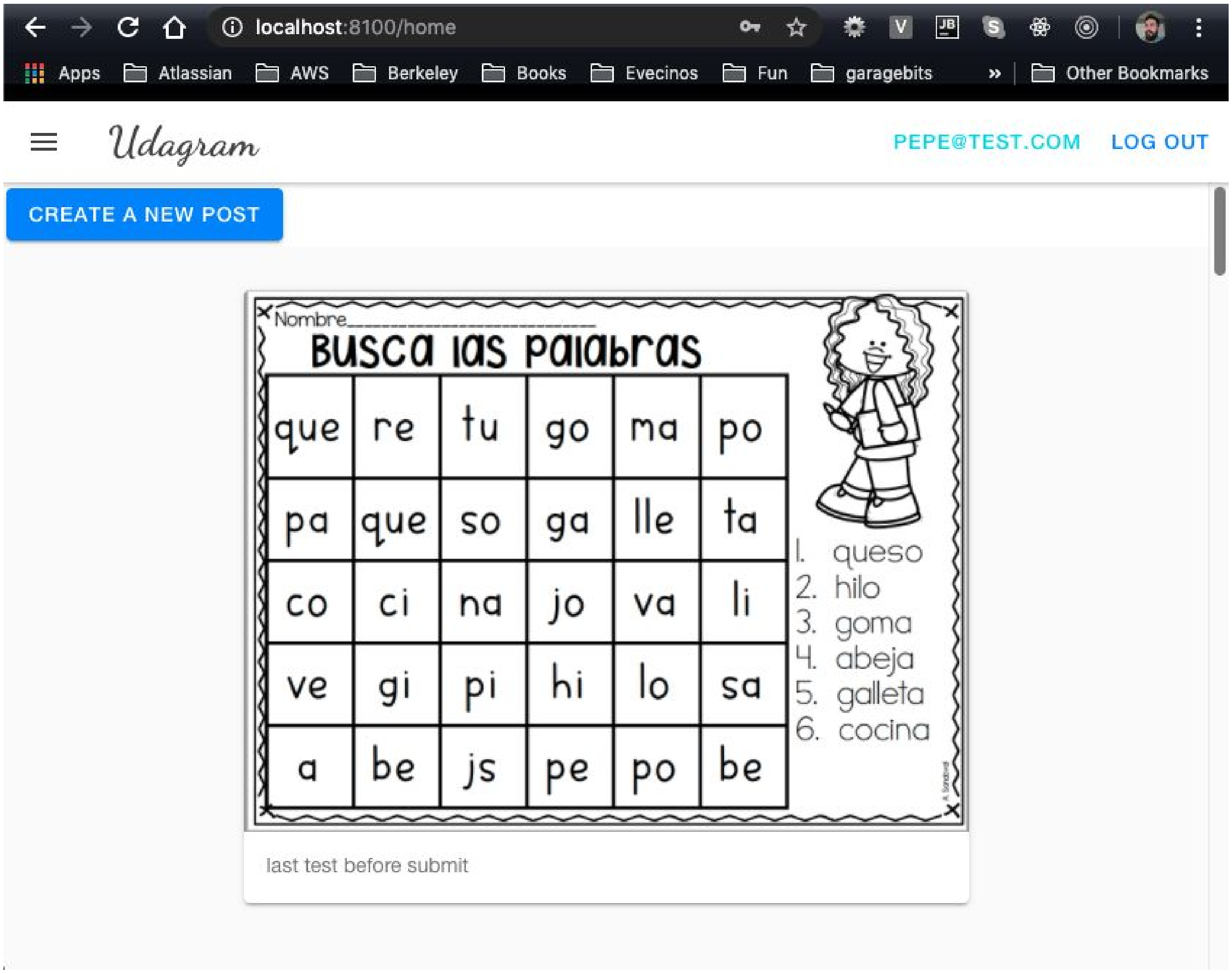


```
tatoo100/reverseproxy (docker_reverseproxy_1 - Up 2 minutes)
tatoo100/udacity-frontend:local (docker_frontend_1 - Up 2 minute...
tatoo100/udacity-restapi-feed (docker_backend-feed_1 - Up 2 mi...
tatoo100/udacity-restapi-user (docker_backend-user_1 - Up 2 mi...
```

```
alpine
beevelop/ionic
docker/desktop-storage-provisioner
docker/kube-compose-api-server
docker/kube-compose-controller
k8s.gcr.io/coredns
k8s.gcr.io/etcd
k8s.gcr.io/kube-apiserver
k8s.gcr.io/kube-controller-manager
k8s.gcr.io/kube-proxy
k8s.gcr.io/kube-scheduler
k8s.gcr.io/pause
nginx
node
tatoo100/reverseproxy
tatoo100/udacity-frontend
tatoo100/udacity-restapi-feed
tatoo100/udacity-restapi-user
```

```yaml
version: "3"
services:
  reverseproxy:
      image: tatoo100/reverseproxy
      ports:
          - 8080:8080
      restart: always
      depends_on:
        - backend-user
        - backend-feed
  backend-user:
    image: tatoo100/udacity-restapi-user
    environment:
      POSTGRES_USERNAME: $POSTGRES_USERNAME
      POSTGRES_PASSWORD: $POSTGRES_PASSWORD
      POSTGRES_DB: $POSTGRES_DB
      POSTGRES_HOST: $POSTGRES_HOST
      AWS_REGION: $AWS_REGION
      AWS_PROFILE: $AWS_PROFILE
      AWS_MEDIA_BUCKET: $AWS_MEDIA_BUCKET
      JWT_SECRET: $JWT_SECRET
      URL: "http://localhost:8100"
  backend-feed:
    image: tatoo100/udacity-restapi-feed
    volumes:
      - $HOME/.aws:/root/.aws
    environment:
      POSTGRES_USERNAME: $POSTGRES_USERNAME
      POSTGRES_PASSWORD: $POSTGRES_PASSWORD
      POSTGRES_DB: $POSTGRES_DB
      POSTGRES_HOST: $POSTGRES_HOST
      AWS_REGION: $AWS_REGION
      AWS_PROFILE: $AWS_PROFILE
      AWS_MEDIA_BUCKET: $AWS_MEDIA_BUCKET
      JWT_SECRET: $JWT_SECRET
      URL: "http://localhost:8100"
  frontend:
    image: tatoo100/udacity-frontend:local
    ports:
      - "8100:80"
```

I was able to login with a user (pepe@test.com)

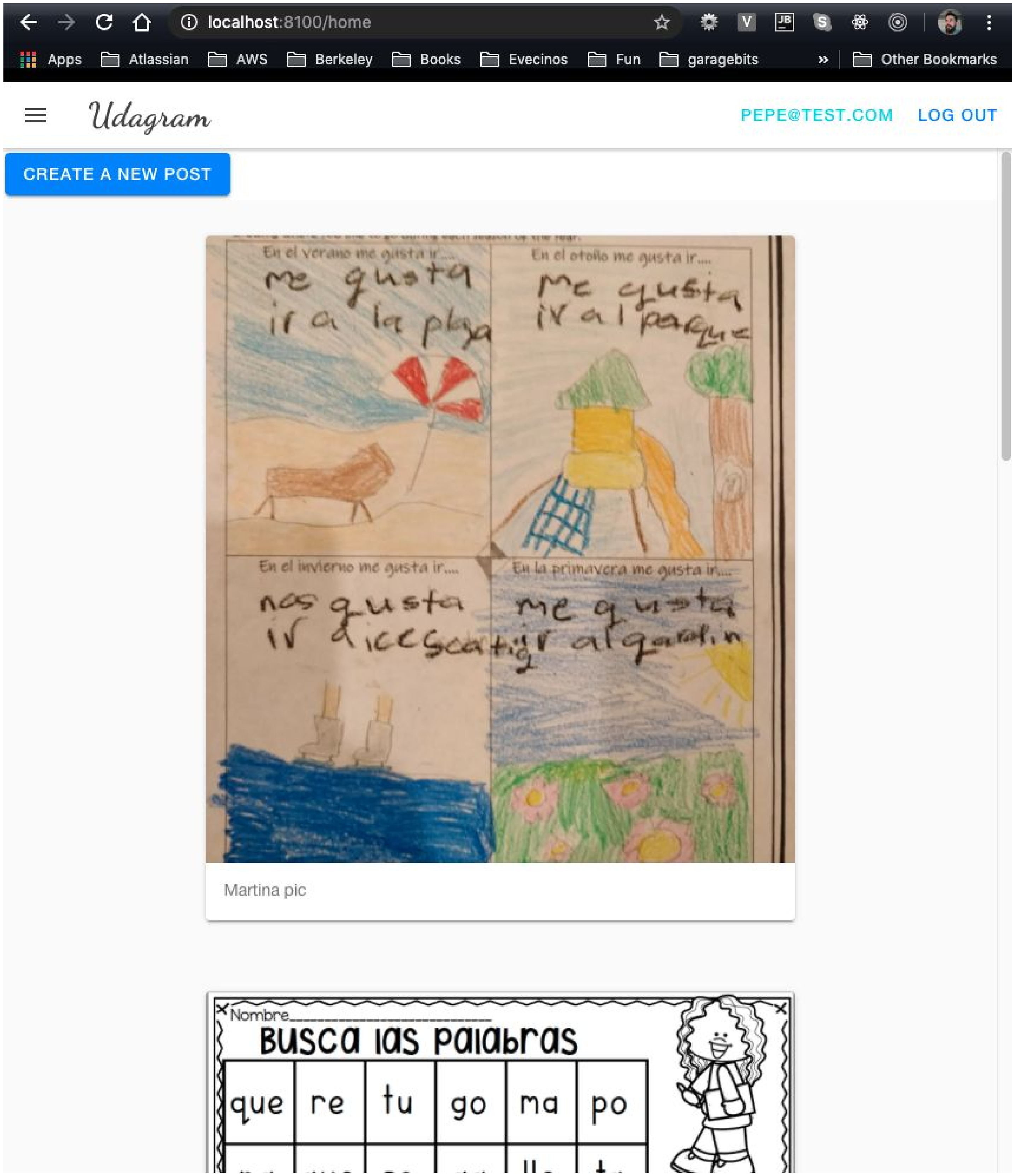


que	re	tu	go	ma	po
pa	que	so	ga	lle	ta
co	ci	na	jo	va	li
ve	gi	pi	hi	lo	sa
a	be	js	pe	po	be

1. queso
2. hilo
3. goma
4. abeja
5. galleta
6. cocina

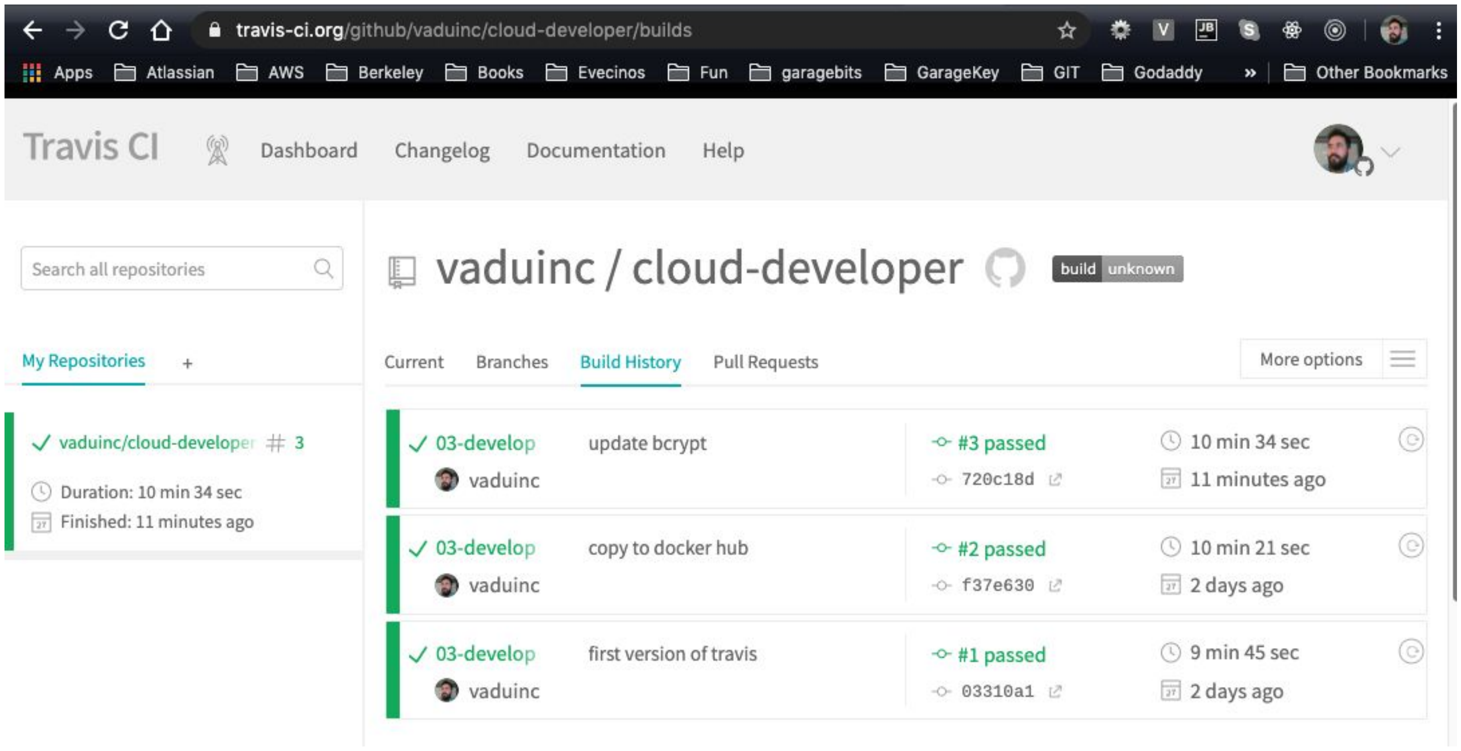
last test before submit

Then I was able to post a new image.



3. Build CICD Pipeline

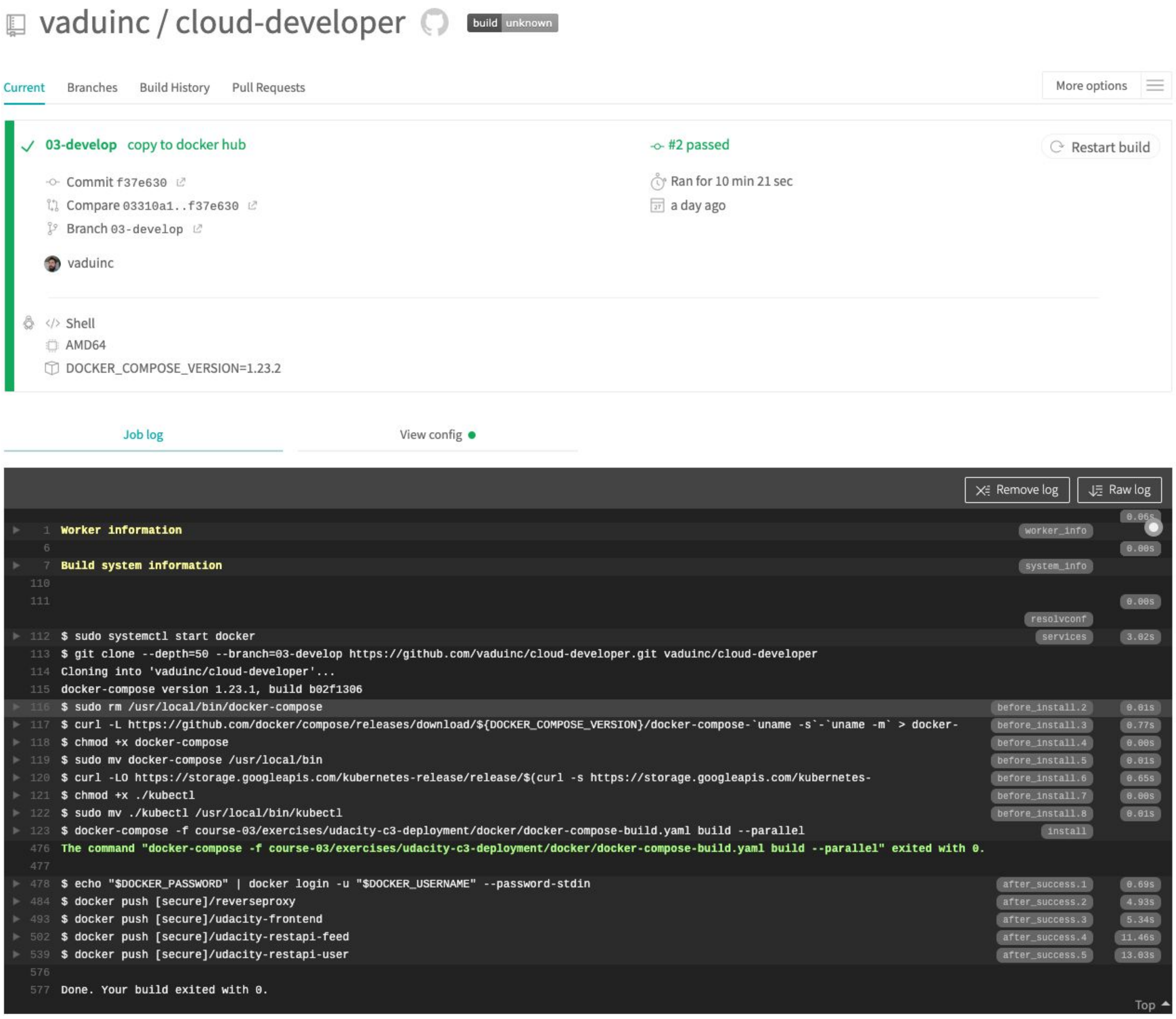
After including .travis.yml file, every push to my GitHub repo triggered a Travis build automatically.



✓ vaduinc/cloud-developer # 3
Duration: 10 min 34 sec
Finished: 11 minutes ago

✓ 03-develop vaduinc	update bcrypt	#3 passed 720c18d	10 min 34 sec 11 minutes ago
✓ 03-develop vaduinc	copy to docker hub	#2 passed f37e630	10 min 21 sec 2 days ago
✓ 03-develop vaduinc	first version of travis	#1 passed 03310a1	9 min 45 sec 2 days ago

The build process finished successfully.



vaduinc / cloud-developer   build | unknown

Current   Branches   Build History   Pull Requests   More options

✓ 03-develop copy to docker hub   #2 passed   Restart build

Commit f37e630   Ran for 10 min 21 sec
Compare 03310a1..f37e630   a day ago
Branch 03-develop

vaduinc

Shell
AMD64
DOCKER_COMPOSE_VERSION=1.23.2

Job log   View config

Remove log   Raw log

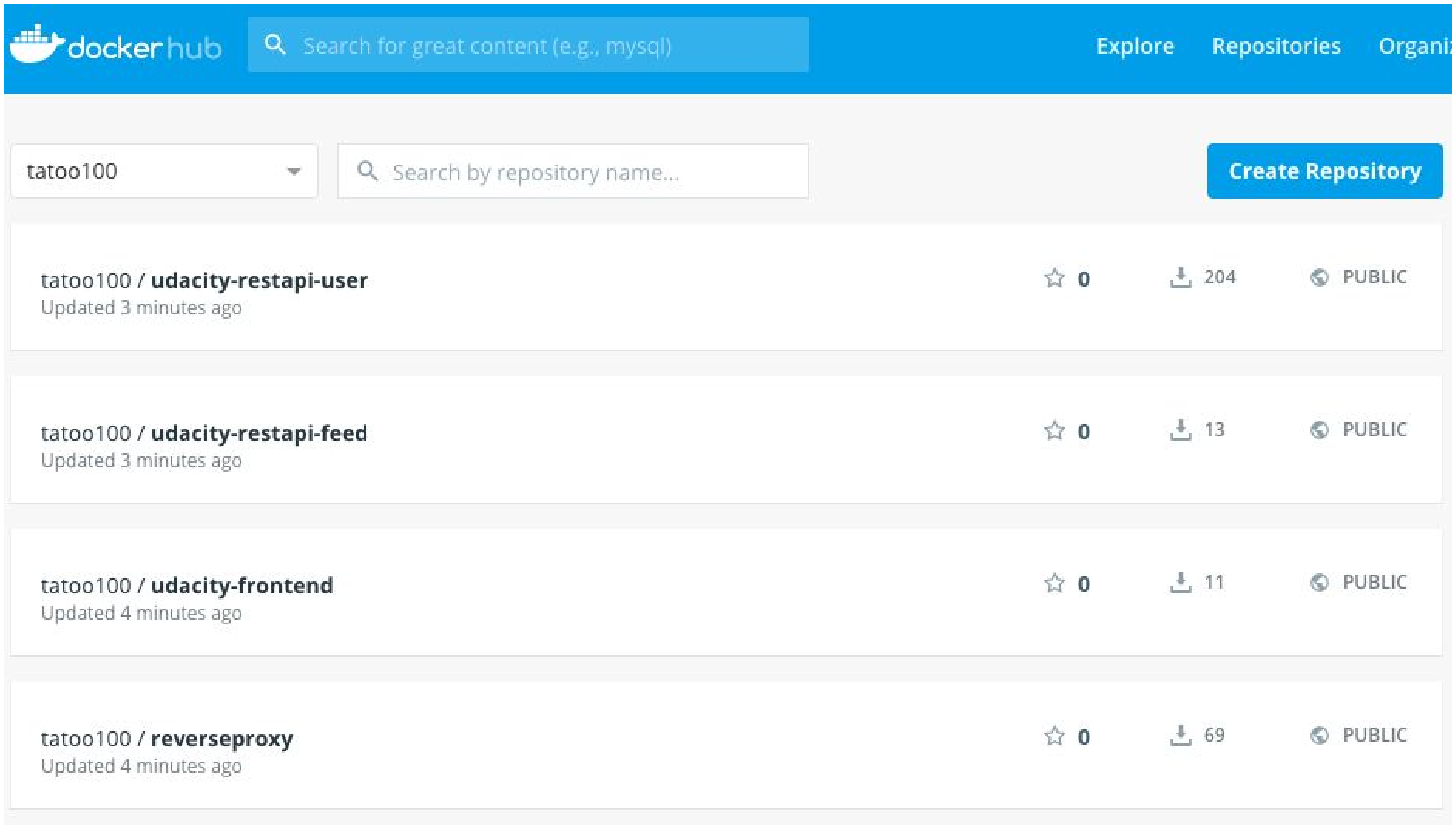
```
Worker information                                                                 worker_info    0.06s
                                                                                                   0.00s
Build system information                                                           system_info
                                                                                                   0.00s
                                                                                   resolvconf
$ sudo systemctl start docker                                                      services       3.02s
$ git clone --depth=50 --branch=03-develop https://github.com/vaduinc/cloud-developer.git vaduinc/cloud-developer
Cloning into 'vaduinc/cloud-developer'...
docker-compose version 1.23.1, build b02f1306
$ sudo rm /usr/local/bin/docker-compose                                            before_install.2   0.01s
$ curl -L https://github.com/docker/compose/releases/download/${DOCKER_COMPOSE_VERSION}/docker-compose-`uname -s`-`uname -m` > docker-   before_install.3   0.77s
$ chmod +x docker-compose                                                          before_install.4   0.00s
$ sudo mv docker-compose /usr/local/bin                                            before_install.5   0.01s
$ curl -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-   before_install.6   0.65s
$ chmod +x ./kubectl                                                               before_install.7   0.00s
$ sudo mv ./kubectl /usr/local/bin/kubectl                                         before_install.8   0.01s
$ docker-compose -f course-03/exercises/udacity-c3-deployment/docker/docker-compose-build.yaml build --parallel   install
The command "docker-compose -f course-03/exercises/udacity-c3-deployment/docker/docker-compose-build.yaml build --parallel" exited with 0.

$ echo "$DOCKER_PASSWORD" | docker login -u "$DOCKER_USERNAME" --password-stdin     after_success.1   0.69s
$ docker push [secure]/reverseproxy                                                after_success.2   4.93s
$ docker push [secure]/udacity-frontend                                            after_success.3   5.34s
$ docker push [secure]/udacity-restapi-feed                                        after_success.4   11.46s
$ docker push [secure]/udacity-restapi-user                                        after_success.5   13.03s

Done. Your build exited with 0.
```

Top

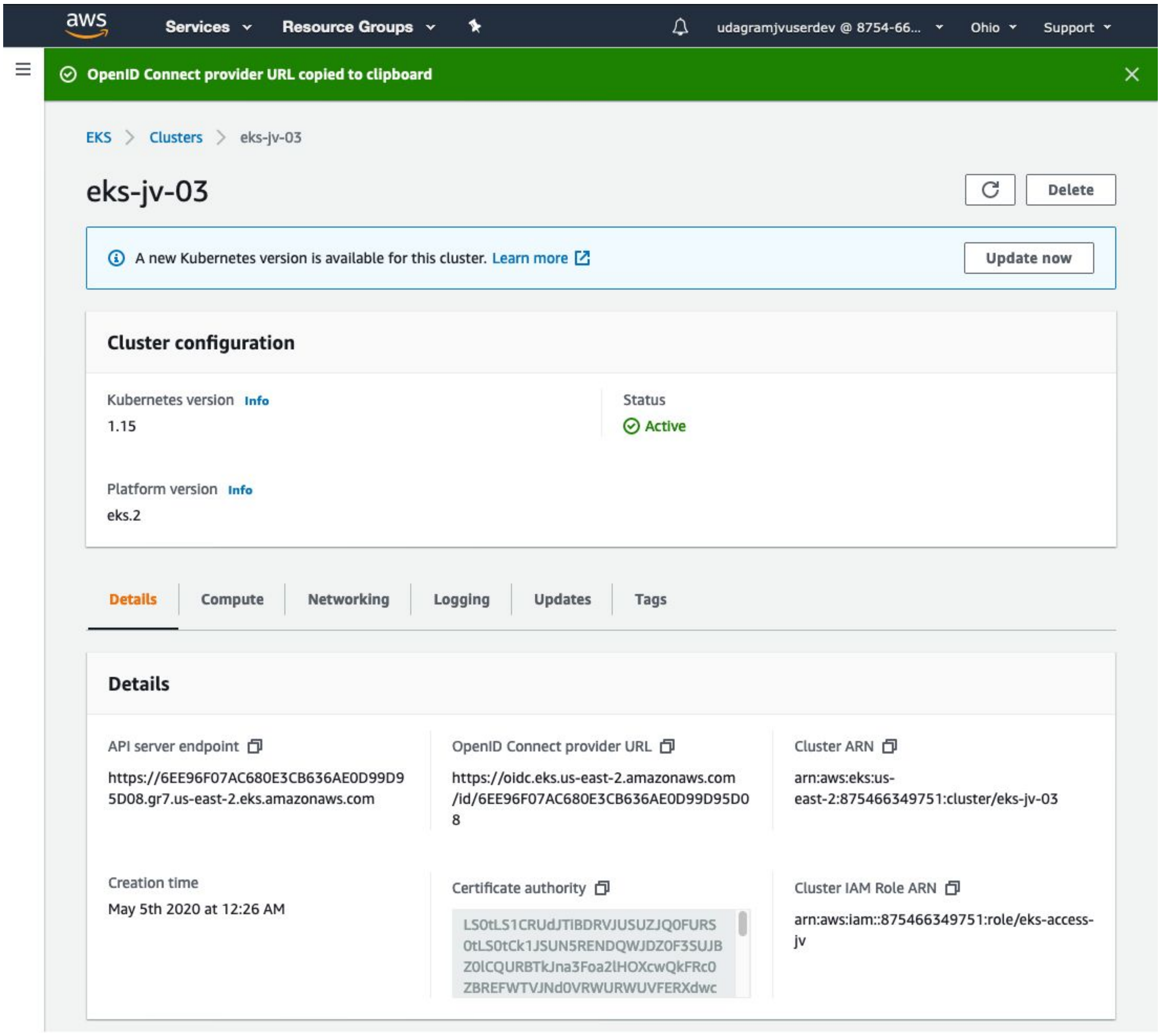
Docker images were pushed to my Docker Hub account.


dockerhub
Search for great content (e.g., mysql)
Explore
Repositories
Organi
tatoo100
Search by repository name...
Create Repository

tatoo100 / **udacity-restapi-user** Updated 3 minutes ago	0	204	PUBLIC
tatoo100 / **udacity-restapi-feed** Updated 3 minutes ago	0	13	PUBLIC
tatoo100 / **udacity-frontend** Updated 4 minutes ago	0	11	PUBLIC
tatoo100 / **reverseproxy** Updated 4 minutes ago	0	69	PUBLIC

4. Deploy to Kubernetes

Created cluster in AWS using EKS. Created the necessary roles for the cluster and nodes. I had to repeat the process and created a second cluster because the user in my local machine was different from the user that created the cluster.



OpenID Connect provider URL copied to clipboard

EKS > Clusters > eks-jv-03

eks-jv-03

Delete

A new Kubernetes version is available for this cluster. Learn more

Update now

Cluster configuration

Kubernetes version Info
1.15

Status
Active

Platform version Info
eks.2

Details | Compute | Networking | Logging | Updates | Tags

Details

API server endpoint
https://6EE96F07AC680E3CB636AE0D99D95D08.gr7.us-east-2.eks.amazonaws.com

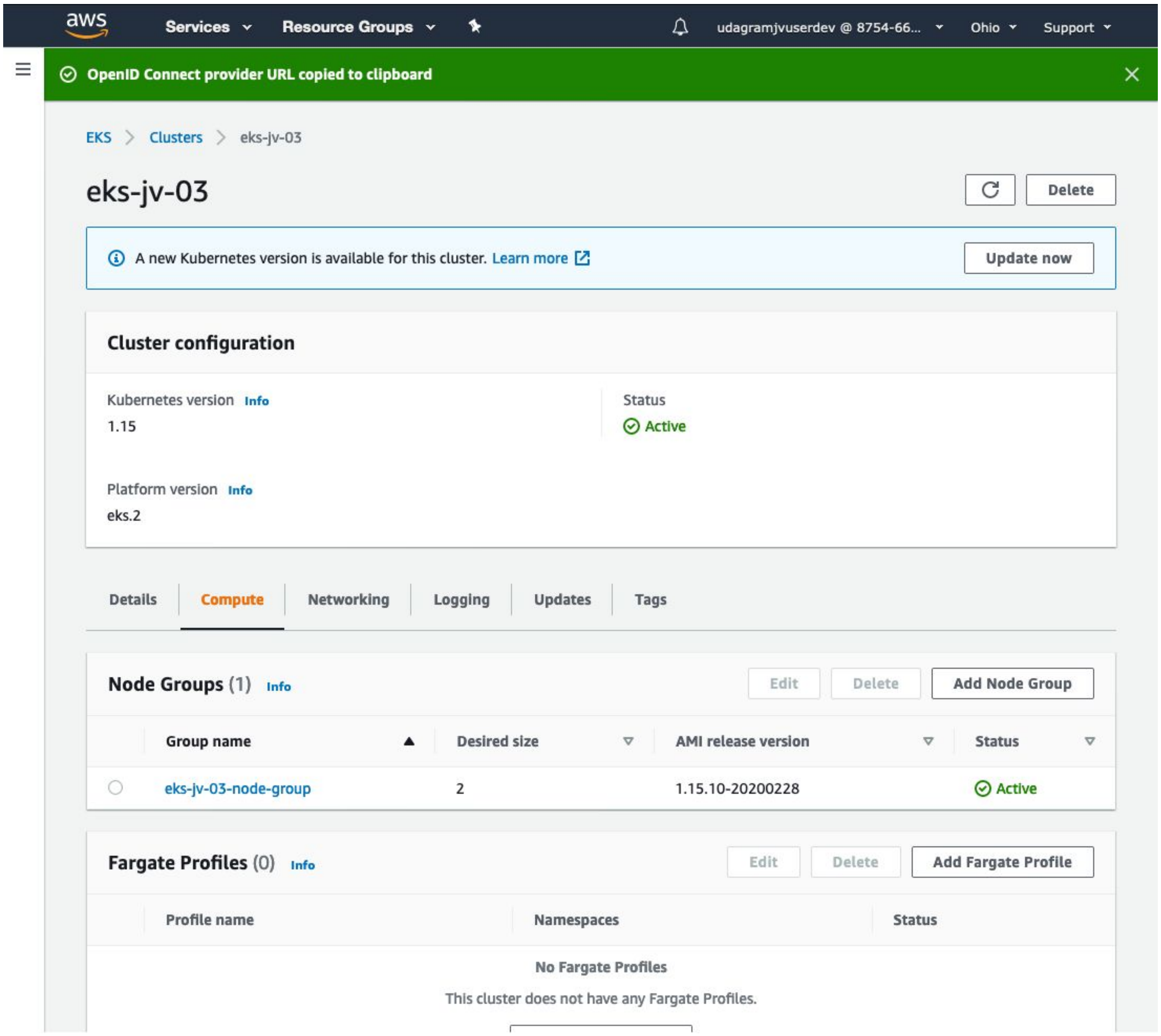
OpenID Connect provider URL
https://oidc.eks.us-east-2.amazonaws.com/id/6EE96F07AC680E3CB636AE0D99D95D08

Cluster ARN
arn:aws:eks:us-east-2:875466349751:cluster/eks-jv-03

Creation time
May 5th 2020 at 12:26 AM

Certificate authority
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUN5RENDQWJDZ0F3SUJBZ0lCQURBTkJna3Foa2lHOXcwQkFRc0ZBREFWTVJNd0VRWURWUVFERXdwc

Cluster IAM Role ARN
arn:aws:iam::875466349751:role/eks-access-jv



OpenID Connect provider URL copied to clipboard

EKS > Clusters > eks-jv-03

eks-jv-03

A new Kubernetes version is available for this cluster. Learn more

Cluster configuration

Kubernetes version Info
1.15

Status
Active

Platform version Info
eks.2

Node Groups (1) Info

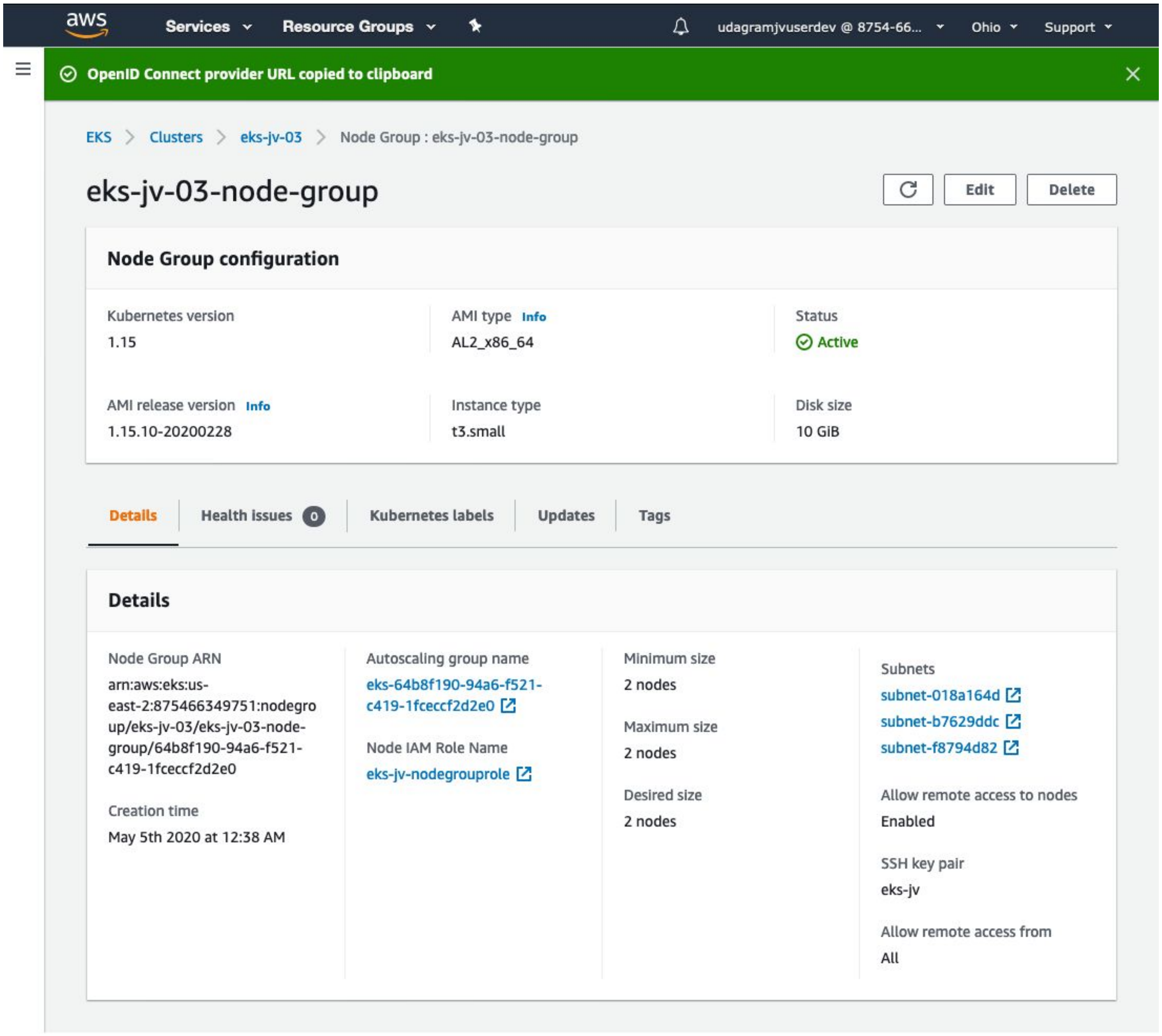
Group name	Desired size	AMI release version	Status
eks-jv-03-node-group	2	1.15.10-20200228	Active

Fargate Profiles (0) Info

Profile name	Namespaces	Status

No Fargate Profiles

This cluster does not have any Fargate Profiles.



OpenID Connect provider URL copied to clipboard

EKS > Clusters > eks-jv-03 > Node Group : eks-jv-03-node-group

eks-jv-03-node-group

Node Group configuration

Kubernetes version 1.15	AMI type Info AL2_x86_64	Status Active
AMI release version Info 1.15.10-20200228	Instance type t3.small	Disk size 10 GiB

Details

Node Group ARN
arn:aws:eks:us-east-2:875466349751:nodegroup/eks-jv-03/eks-jv-03-node-group/64b8f190-94a6-f521-c419-1fceccf2d2e0

Creation time
May 5th 2020 at 12:38 AM

Autoscaling group name
eks-64b8f190-94a6-f521-c419-1fceccf2d2e0

Node IAM Role Name
eks-jv-nodegrouprole

Minimum size
2 nodes

Maximum size
2 nodes

Desired size
2 nodes

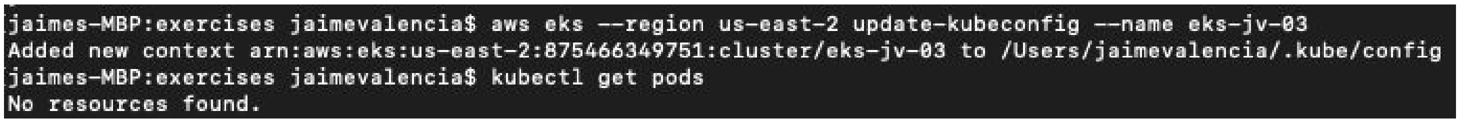
Subnets
subnet-018a164d
subnet-b7629ddc
subnet-f8794d82

Allow remote access to nodes
Enabled

SSH key pair
eks-jv

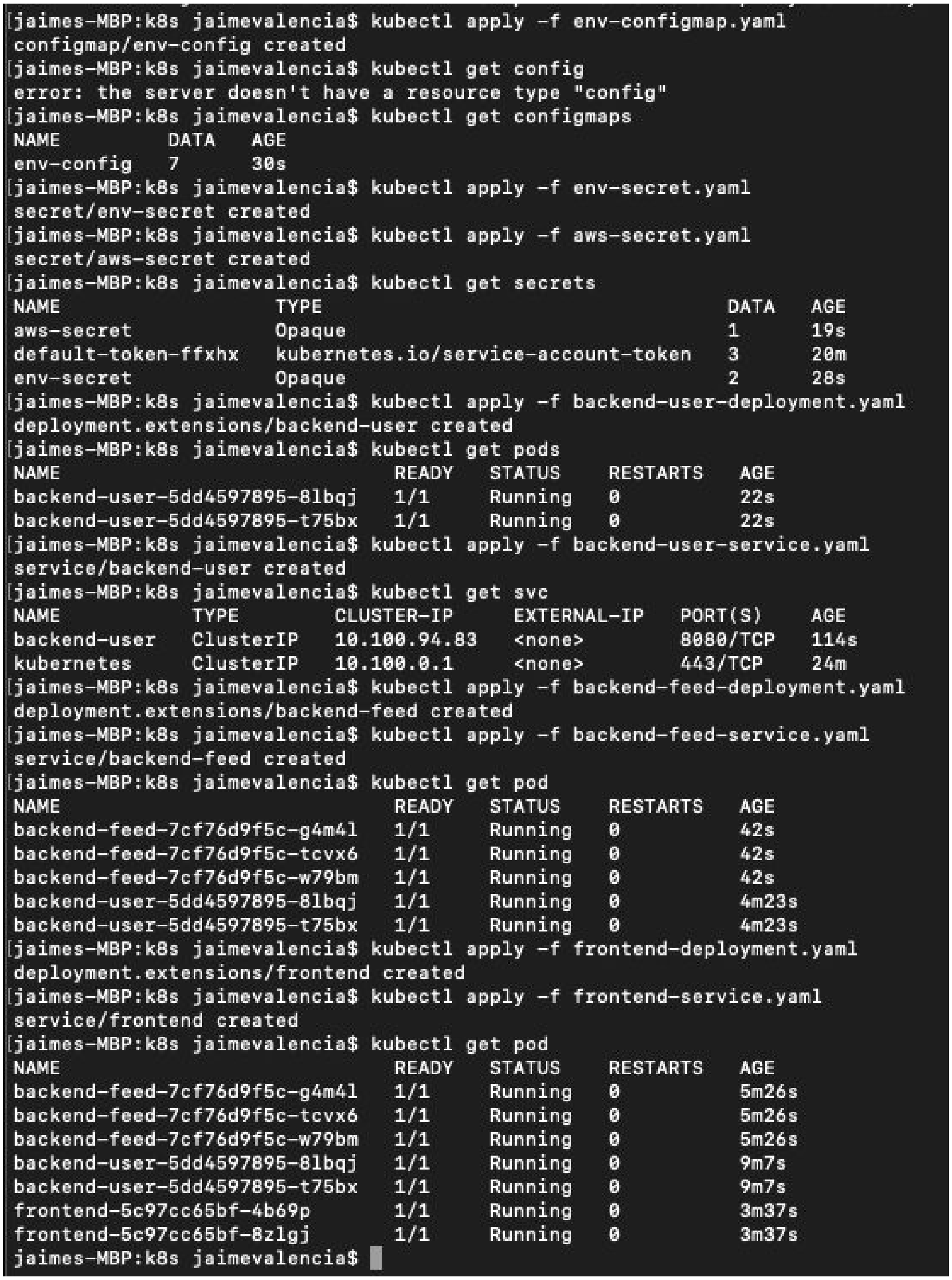
Allow remote access from
All

Followed the instructions to attached my local computer kubectl CLI to the previous created EKS cluster in AWS



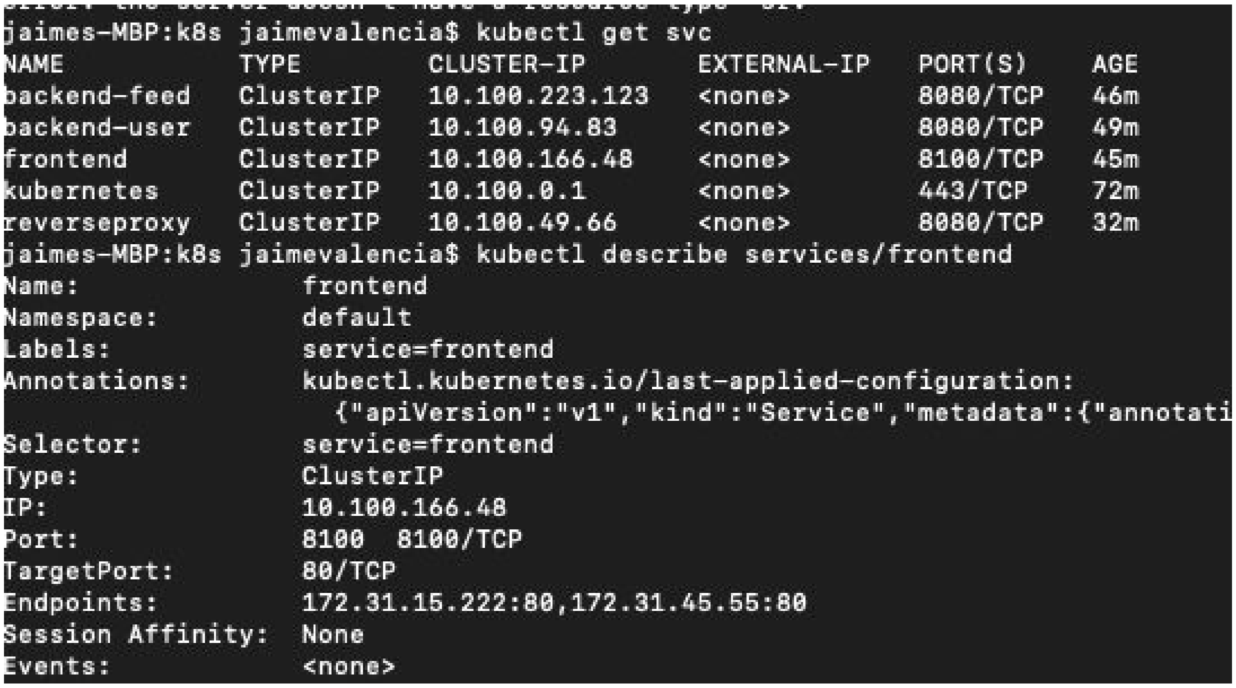
```
jaimes-MBP:exercises jaimevalencia$ aws eks --region us-east-2 update-kubeconfig --name eks-jv-03
Added new context arn:aws:eks:us-east-2:875466349751:cluster/eks-jv-03 to /Users/jaimevalencia/.kube/config
jaimes-MBP:exercises jaimevalencia$ kubectl get pods
No resources found.
```

Then started applying/creating each of the yaml files, including configuration, secrets, deployments and services.



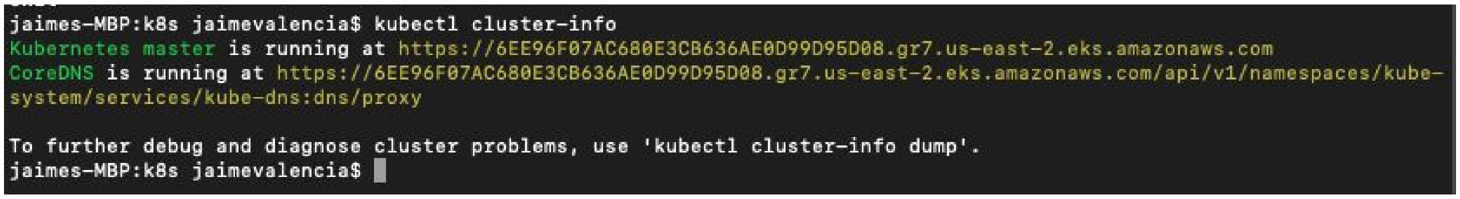
```
[jaimes-MBP:k8s jaimevalencia$ kubectl apply -f env-configmap.yaml
configmap/env-config created
[jaimes-MBP:k8s jaimevalencia$ kubectl get config
error: the server doesn't have a resource type "config"
[jaimes-MBP:k8s jaimevalencia$ kubectl get configmaps
NAME         DATA   AGE
env-config   7      30s
[jaimes-MBP:k8s jaimevalencia$ kubectl apply -f env-secret.yaml
secret/env-secret created
[jaimes-MBP:k8s jaimevalencia$ kubectl apply -f aws-secret.yaml
secret/aws-secret created
[jaimes-MBP:k8s jaimevalencia$ kubectl get secrets
NAME                  TYPE                                  DATA   AGE
aws-secret            Opaque                                1      19s
default-token-ffxhx   kubernetes.io/service-account-token   3      20m
env-secret            Opaque                                2      28s
[jaimes-MBP:k8s jaimevalencia$ kubectl apply -f backend-user-deployment.yaml
deployment.extensions/backend-user created
[jaimes-MBP:k8s jaimevalencia$ kubectl get pods
NAME                            READY   STATUS    RESTARTS   AGE
backend-user-5dd4597895-8lbqj   1/1     Running   0          22s
backend-user-5dd4597895-t75bx   1/1     Running   0          22s
[jaimes-MBP:k8s jaimevalencia$ kubectl apply -f backend-user-service.yaml
service/backend-user created
[jaimes-MBP:k8s jaimevalencia$ kubectl get svc
NAME           TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)    AGE
backend-user   ClusterIP   10.100.94.83   <none>        8080/TCP   114s
kubernetes     ClusterIP   10.100.0.1     <none>        443/TCP    24m
[jaimes-MBP:k8s jaimevalencia$ kubectl apply -f backend-feed-deployment.yaml
deployment.extensions/backend-feed created
[jaimes-MBP:k8s jaimevalencia$ kubectl apply -f backend-feed-service.yaml
service/backend-feed created
[jaimes-MBP:k8s jaimevalencia$ kubectl get pod
NAME                            READY   STATUS    RESTARTS   AGE
backend-feed-7cf76d9f5c-g4m4l   1/1     Running   0          42s
backend-feed-7cf76d9f5c-tcvx6   1/1     Running   0          42s
backend-feed-7cf76d9f5c-w79bm   1/1     Running   0          42s
backend-user-5dd4597895-8lbqj   1/1     Running   0          4m23s
backend-user-5dd4597895-t75bx   1/1     Running   0          4m23s
[jaimes-MBP:k8s jaimevalencia$ kubectl apply -f frontend-deployment.yaml
deployment.extensions/frontend created
[jaimes-MBP:k8s jaimevalencia$ kubectl apply -f frontend-service.yaml
service/frontend created
[jaimes-MBP:k8s jaimevalencia$ kubectl get pod
NAME                            READY   STATUS    RESTARTS   AGE
backend-feed-7cf76d9f5c-g4m4l   1/1     Running   0          5m26s
backend-feed-7cf76d9f5c-tcvx6   1/1     Running   0          5m26s
backend-feed-7cf76d9f5c-w79bm   1/1     Running   0          5m26s
backend-user-5dd4597895-8lbqj   1/1     Running   0          9m7s
backend-user-5dd4597895-t75bx   1/1     Running   0          9m7s
frontend-5c97cc65bf-4b69p       1/1     Running   0          3m37s
frontend-5c97cc65bf-8zlgj       1/1     Running   0          3m37s
jaimes-MBP:k8s jaimevalencia$
```

Execute command; `kubectl describe services/frontend`



```
jaimes-MBP:k8s jaimevalencia$ kubectl get svc
NAME            TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)    AGE
backend-feed    ClusterIP   10.100.223.123   <none>        8080/TCP   46m
backend-user    ClusterIP   10.100.94.83     <none>        8080/TCP   49m
frontend        ClusterIP   10.100.166.48    <none>        8100/TCP   45m
kubernetes      ClusterIP   10.100.0.1       <none>        443/TCP    72m
reverseproxy    ClusterIP   10.100.49.66     <none>        8080/TCP   32m
jaimes-MBP:k8s jaimevalencia$ kubectl describe services/frontend
Name:              frontend
Namespace:         default
Labels:            service=frontend
Annotations:       kubectl.kubernetes.io/last-applied-configuration:
                     {"apiVersion":"v1","kind":"Service","metadata":{"annotati
Selector:          service=frontend
Type:              ClusterIP
IP:                10.100.166.48
Port:              8100  8100/TCP
TargetPort:        80/TCP
Endpoints:         172.31.15.222:80,172.31.45.55:80
Session Affinity:  None
Events:            <none>
```

Some details about the cluster after executing command: `kubectl cluster-info`



```
jaimes-MBP:k8s jaimevalencia$ kubectl cluster-info
Kubernetes master is running at https://6EE96F07AC680E3CB636AE0D99D95D08.gr7.us-east-2.eks.amazonaws.com
CoreDNS is running at https://6EE96F07AC680E3CB636AE0D99D95D08.gr7.us-east-2.eks.amazonaws.com/api/v1/namespaces/kube-
system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
jaimes-MBP:k8s jaimevalencia$
```