**CI/DC, Github & Code Quality**

1. All project code is stored in a GitHub repository and a link to the repository has been provided for reviewers. The student uses a CI tool to build the application.

1.1 Github repository:<https://github.com/stonexjr/cloud-developer/tree/06-ci/course-03/exercises>

1.2 Screenshot of TravisCI which shows the successful build and deploy steps

The image is too big to show in the document, and is available on the github repo: <https://github.com/stonexjr/cloud-developer/blob/06-ci/course-03/submission/travis-ci-screenshot.png>

The .travis.yml at the root of the repository has been revised to allow the build images being pushed to Dockerhub so that it can be pulled by kubectl on remote eks clusters.

2. The README file includes introduction how to setup and deploy the project. It explains the main building blocks and has comments in the important files.

README: <https://github.com/stonexjr/cloud-developer/blob/06-ci/README.md>

**Container**

1. The app is containerized. There is a Dockerfile in repo and the docker image can be build

There is a Dockerfile available in each subfolder of the three services:

**Backend-user Dockerfile**: <https://github.com/stonexjr/cloud-developer/tree/06-ci/course-03/exercises/udacity-c3-restapi-feed>

**Backend-feed Dockerfile**: <https://github.com/stonexjr/cloud-developer/tree/06-ci/course-03/exercises/udacity-c3-restapi-user>

**Frontend Dockerfile**: <https://github.com/stonexjr/cloud-developer/tree/06-ci/course-03/exercises/udacity-c3-frontend>

**Reverseproxy** **Dockerfile**: <https://github.com/stonexjr/cloud-developer/tree/06-ci/course-03/exercises/udacity-c3-deployment/docker>

2. The project have public docker images.

The DockerHub images for the application are available here:

<https://hub.docker.com/repository/docker/stonexjr/reverseproxy>

<https://hub.docker.com/repository/docker/stonexjr/udacity-frontend>

<https://hub.docker.com/repository/docker/stonexjr/udacity-restapi-feed>

<https://hub.docker.com/repository/docker/stonexjr/udacity-restapi-user>

3. The applications runs in a container without errors.

Starting the app as a container on a local system by executing

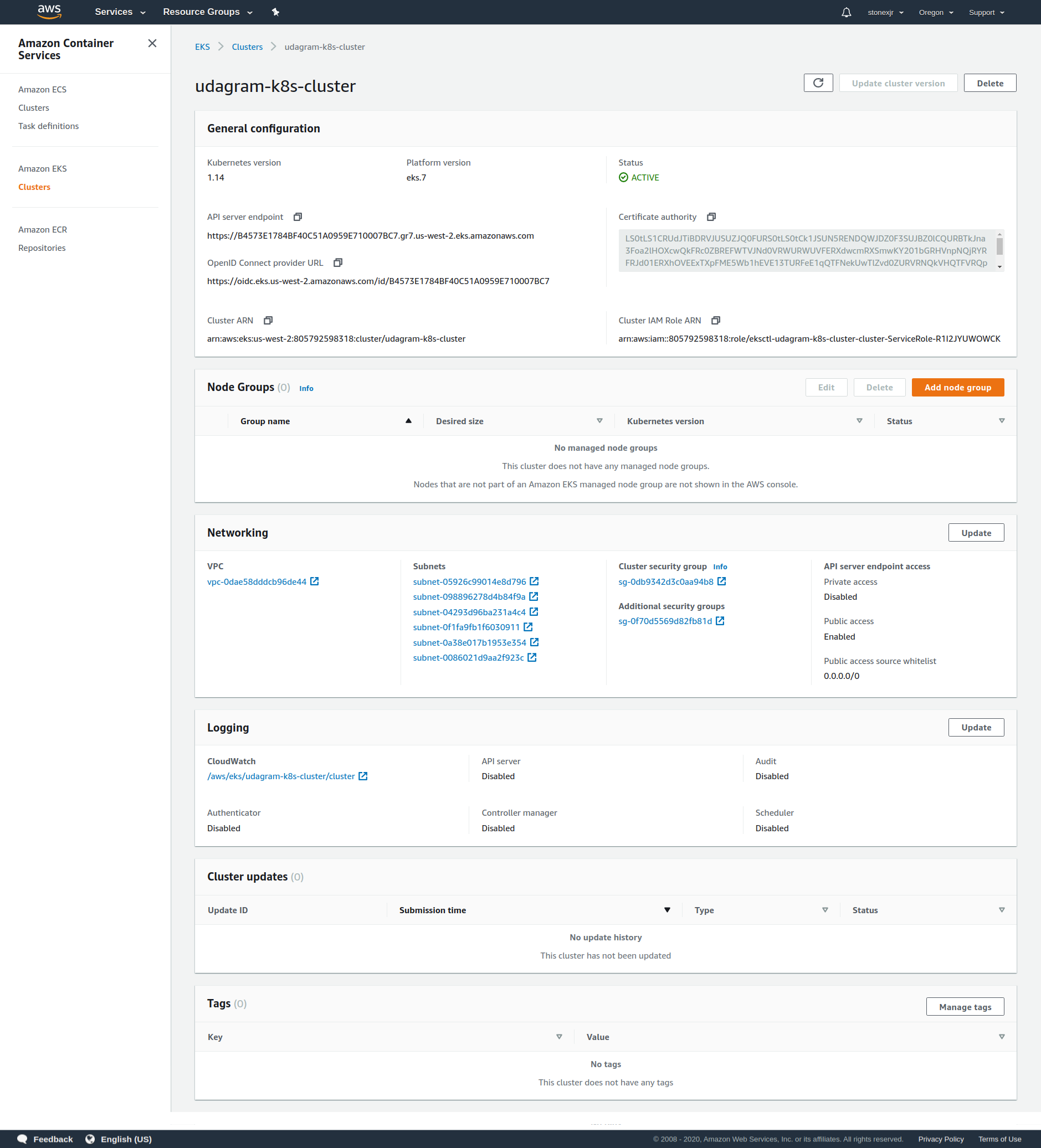
$>cd course-03/exercises/udacity-c3-deployment/docker

$>docker-compose up

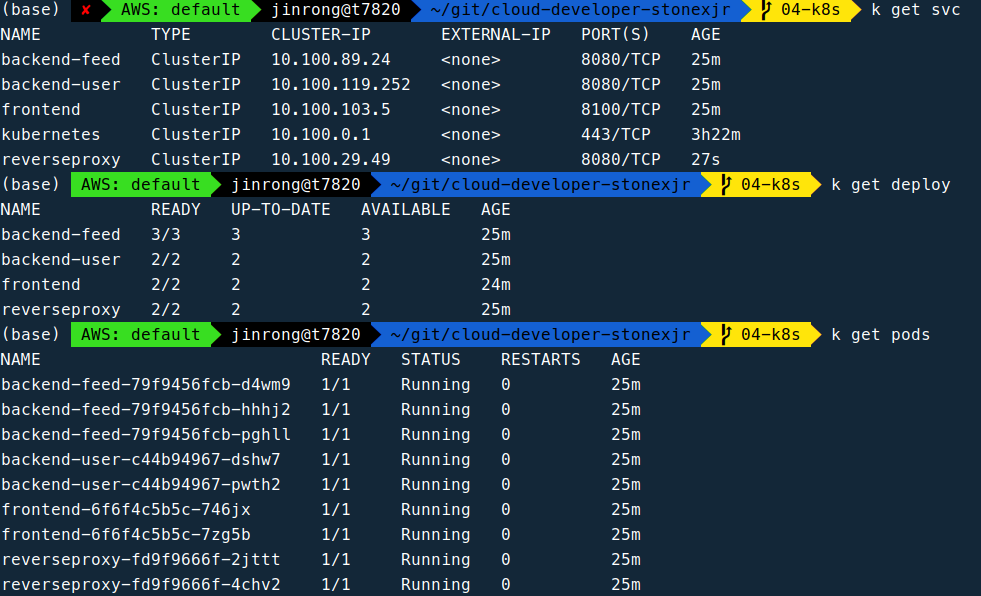
**Deployment**

1. The application runs on a cluster in the cloud. The project can be deployed to a kubernetes cluster

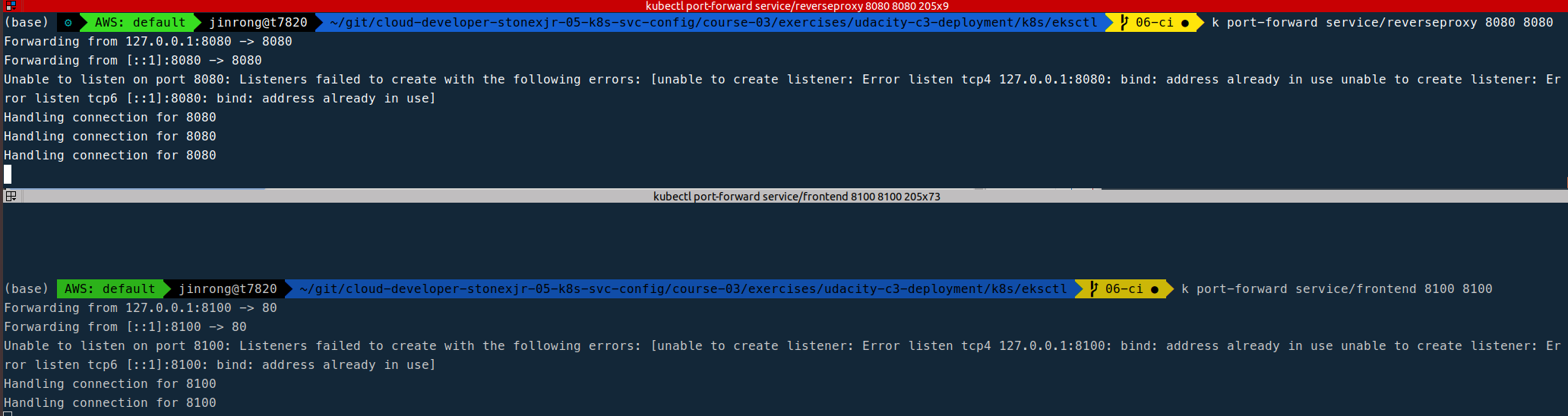
Here are the screenshots of provisioned AWS EKS clusters



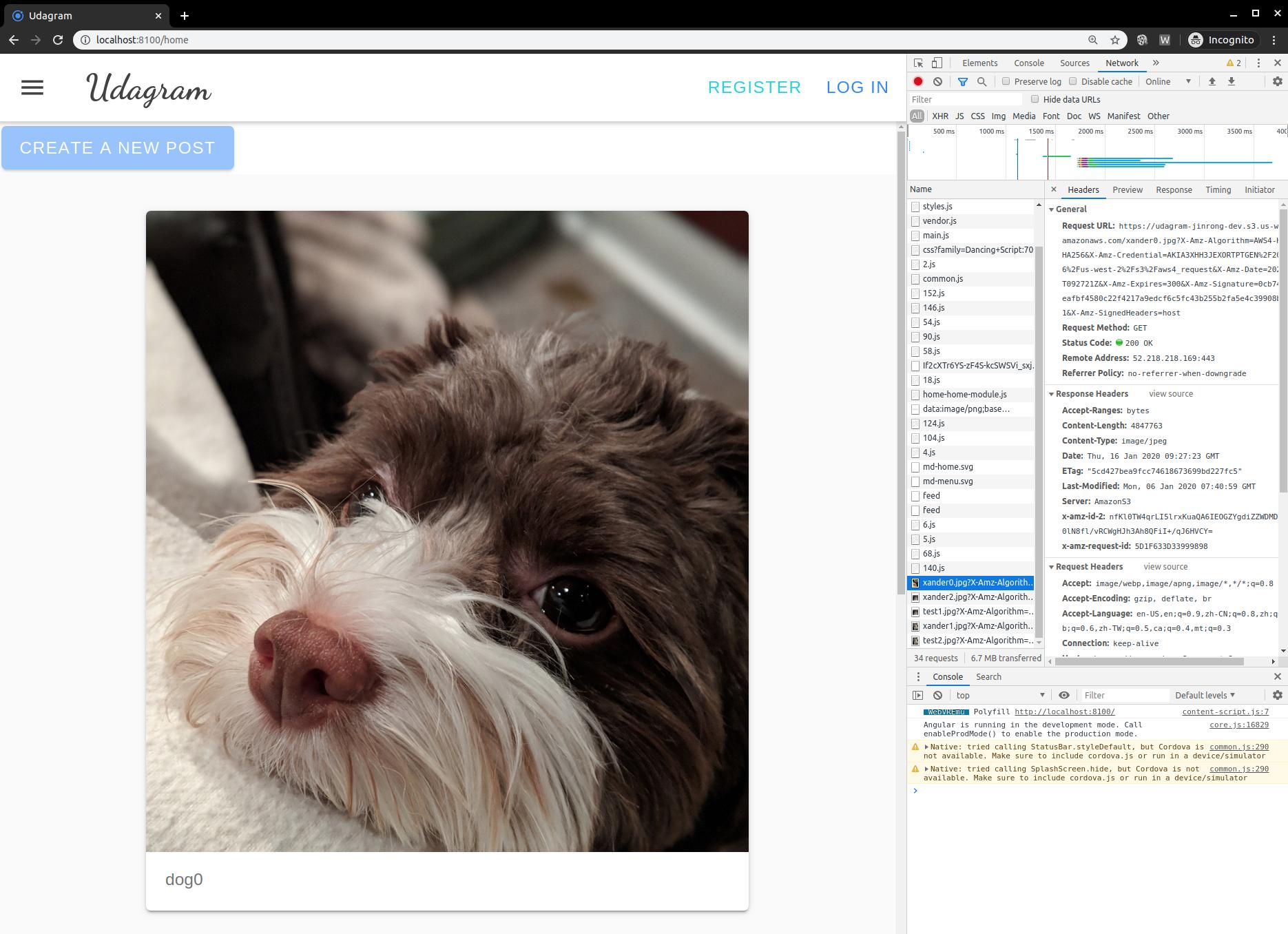
And the following is a screenshot of information on the deployed services, deployments and pods



2. Using kubectl port-forward to access both frontend and backend service through localhost at port 8100 and 8080 respectively. See screenshot below



3. Frontend website screen shot



4. CloudWatch Mointoring of k8s ec2 nodes and autoscaling group

