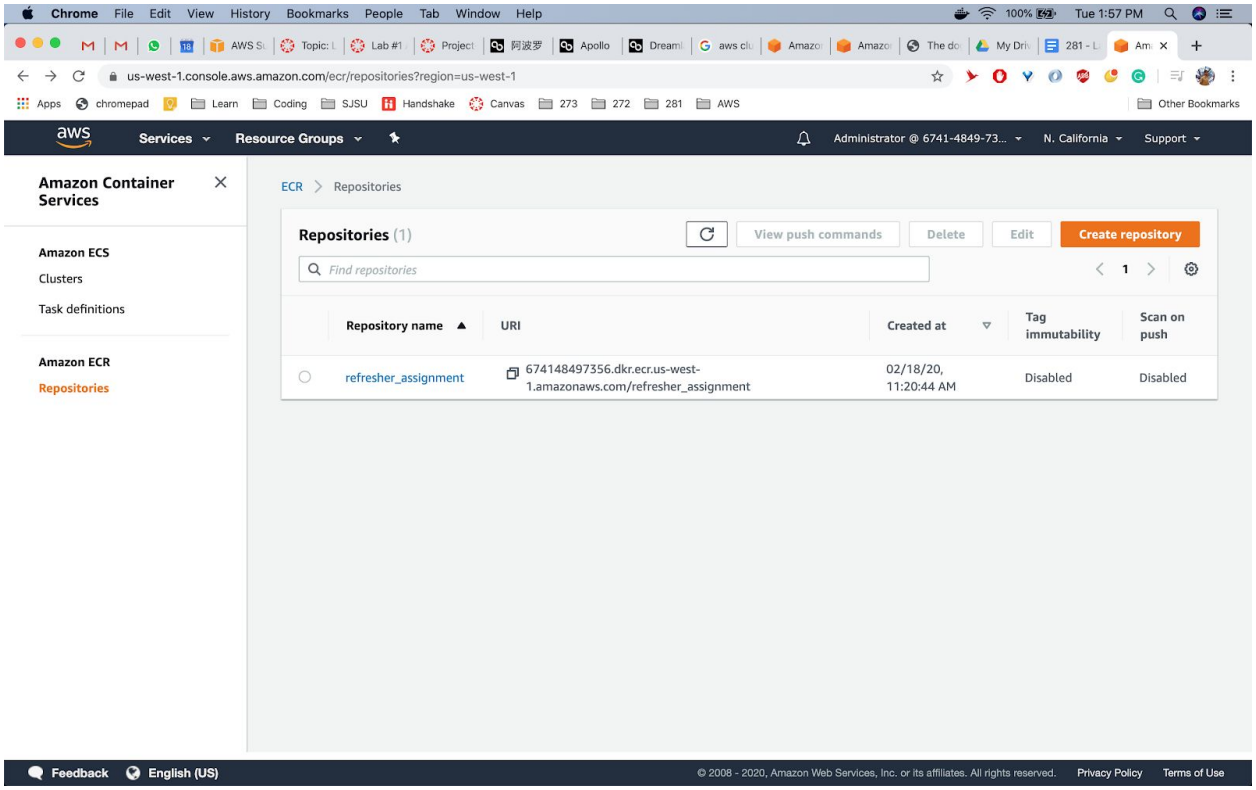
Lab #1 Assignment - Amazon EC2

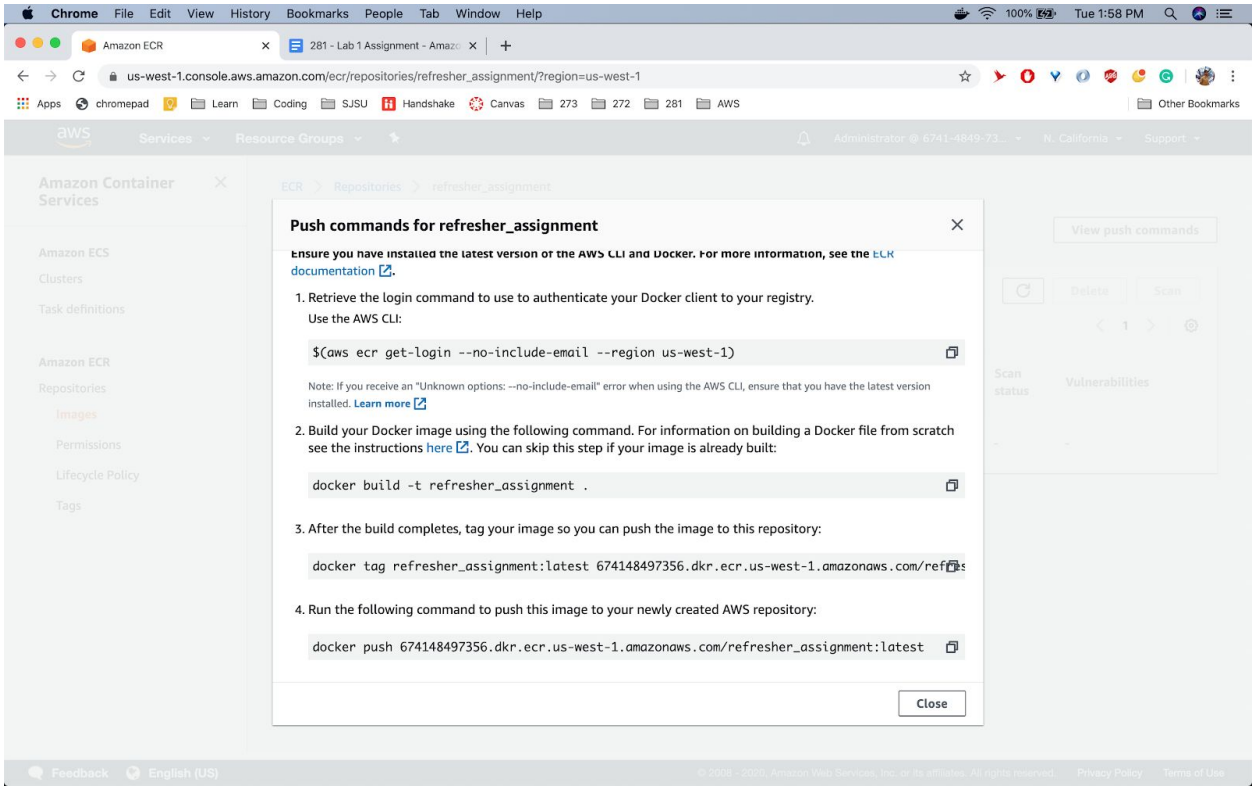
Here I will be describing how I created a simple web application and deployed it in AWS EC2 using docker and ECS.

First I created a simple web app in node js which will display a random list of dog breeds and shows info and picture of the breed clicked

After creating the application, I built the application using docker. A docker image got created Next, I created an account in AWS and created a Amazon Elastic Container Registry(ECR) which is used to store and maintain docker images, similar to docker hub.

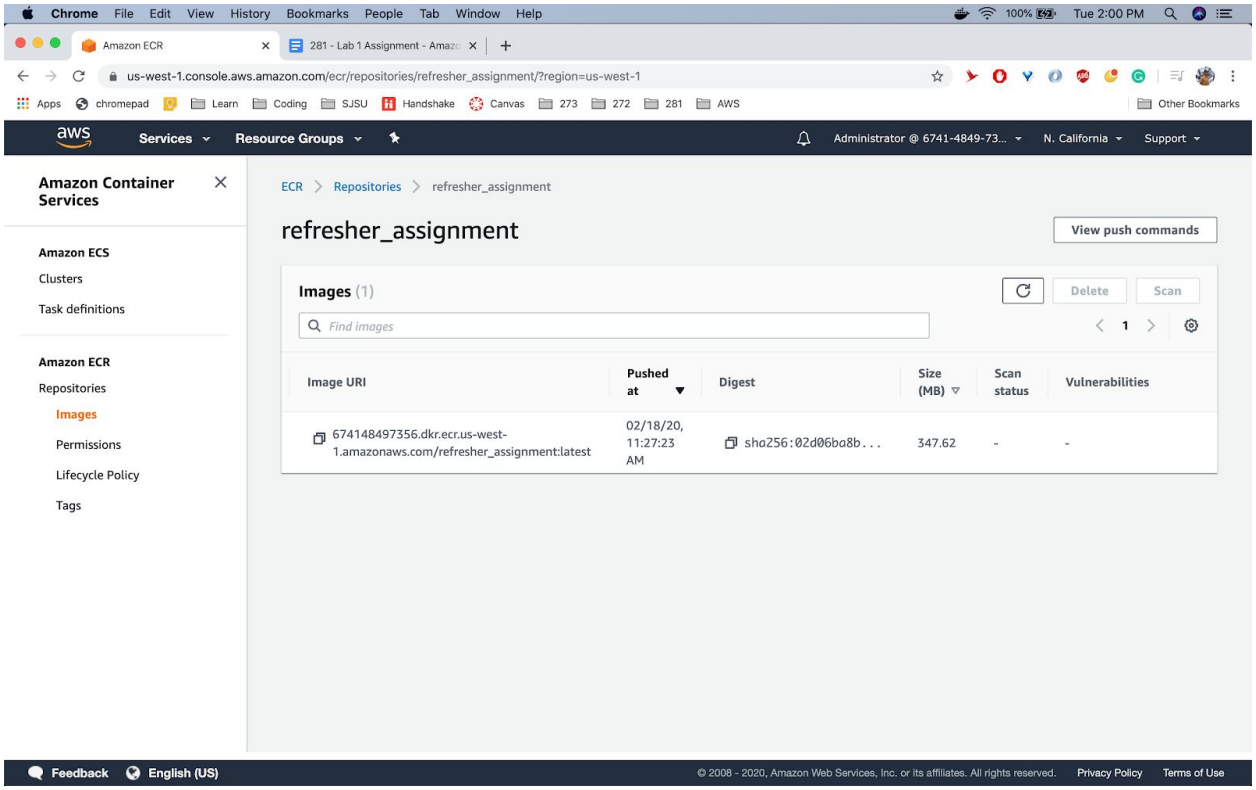


Then followed the below command



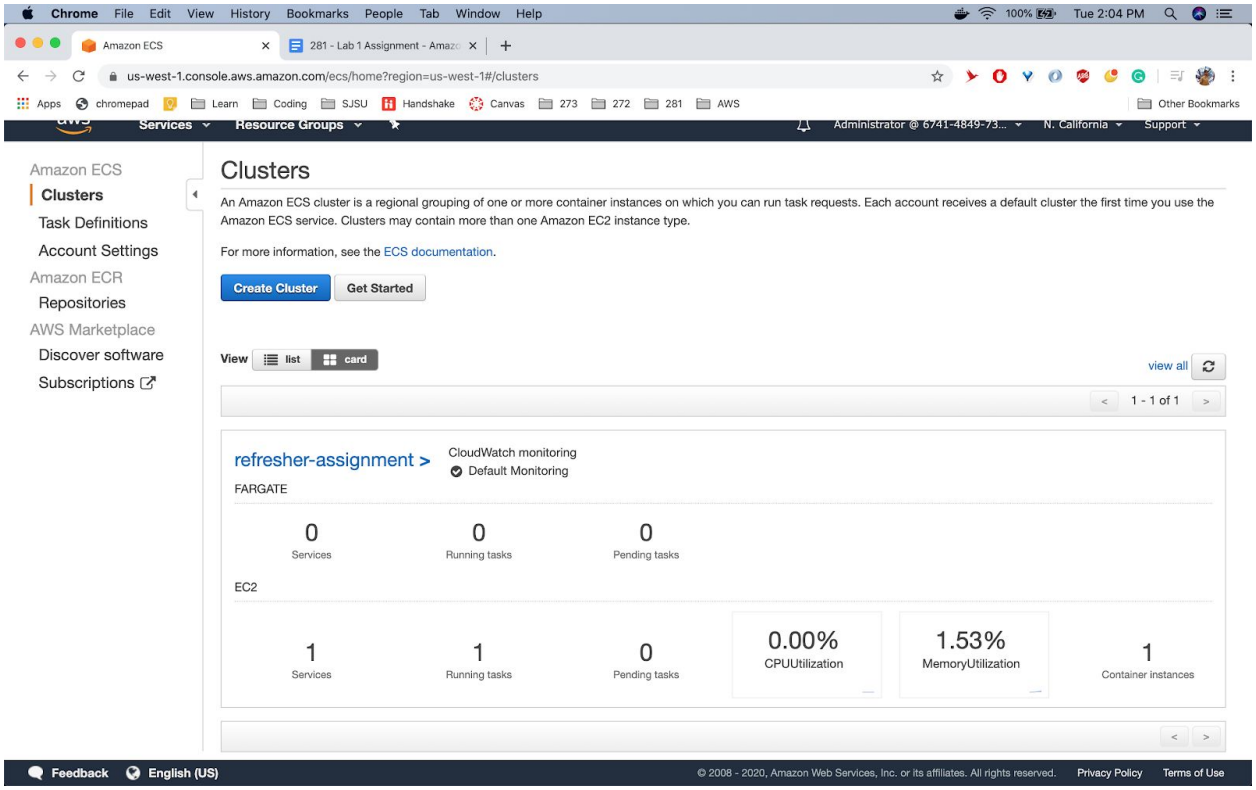
Which created an image and pushed it to the ECR

```
prashanths-MacBook-Air:app prashanthn$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
prashanths-MacBook-Air:app prashanthn$ docker images
REPOSITORY                                                      TAG                 IMAGE ID            CREATED             SIZE
674148497356.dkr.ecr.us-west-1.amazonaws.com/refresher_assignment   latest              7bb03d03a8d1        2 hours ago         902MB
refresher_assignment                                            latest              7bb03d03a8d1        2 hours ago         902MB
674148497356.dkr.ecr.us-east-2.amazonaws.com/refresher_assignment   latest              3810e40f8f7e        10 days ago         904MB
<none>                                                          <none>              1b9c44f2d163        11 days ago         902MB
<none>                                                          <none>              da98cffcd684        11 days ago         902MB
<none>                                                          <none>              a6d926c10872        11 days ago         902MB
<none>                                                          <none>              e73f3e3492e2        11 days ago         902MB
<none>                                                          <none>              5dae38d1b9a0        11 days ago         902MB
<none>                                                          <none>              7e86b256ae76        11 days ago         902MB
<none>                                                          <none>              3e3b6977376e        11 days ago         902MB
<none>                                                          <none>              1d95d218b34b        11 days ago         902MB
<none>                                                          <none>              3ecb931951c3        11 days ago         902MB
<none>                                                          <none>              1053cb53cf8a        11 days ago         902MB
<none>                                                          <none>              e57930026d32        11 days ago         902MB
<none>                                                          <none>              c13847acccec        11 days ago         928MB
<none>                                                          <none>              9b043d6a7974        11 days ago         928MB
<none>                                                          <none>              76fa01390738        11 days ago         928MB
<none>                                                          <none>              2b42955403ff        11 days ago         928MB
<none>                                                          <none>              f3c47eb8af7b        11 days ago         928MB
<none>                                                          <none>              dc2b9134b9ea        11 days ago         928MB
<none>                                                          <none>              99ed8815af71        11 days ago         928MB
<none>                                                          <none>              9a9a22e8ac39        11 days ago         928MB
<none>                                                          <none>              fe58fe62e202        11 days ago         928MB
<none>                                                          <none>              c314fedb1c5d        11 days ago         928MB
<none>                                                          <none>              a288aa9549cf        11 days ago         928MB
<none>                                                          <none>              f800ec7163f8        11 days ago         928MB
<none>                                                          <none>              5efcf5367e5a        11 days ago         928MB
node_app                                                        v0.0.2              92e218e43f58        12 days ago         907MB
node_app                                                        v0.0.1              71a3488d2160        12 days ago         907MB
<none>                                                          <none>              e037275b3f67        12 days ago         907MB
<none>                                                          <none>              c55391bc9988        12 days ago         907MB
alpine                                                          latest              e7d92cdc71fe        4 weeks ago         5.59MB
node                                                            10.15.1             8fc2110c6978        12 months ago       897MB
prashanths-MacBook-Air:app prashanthn$
```



To deploy to web,
First I created a ECS cluster with the following steps

1. Selected EC2 Linux + Networking
2. Gave the cluster name - refresher-assignment
3. Selected the free tier instance - t2.micro
4. Number of instances - 1
5. EBS storage (GiB) - 22
6. Left rest of the configurations as default



Next created a new Task definition with the following steps

1. Selected ec2 as the launch type
2. Gave Task Definition Name
3. Added task memory and CPU units
4. Added container details

Below is the JSON definition of the task

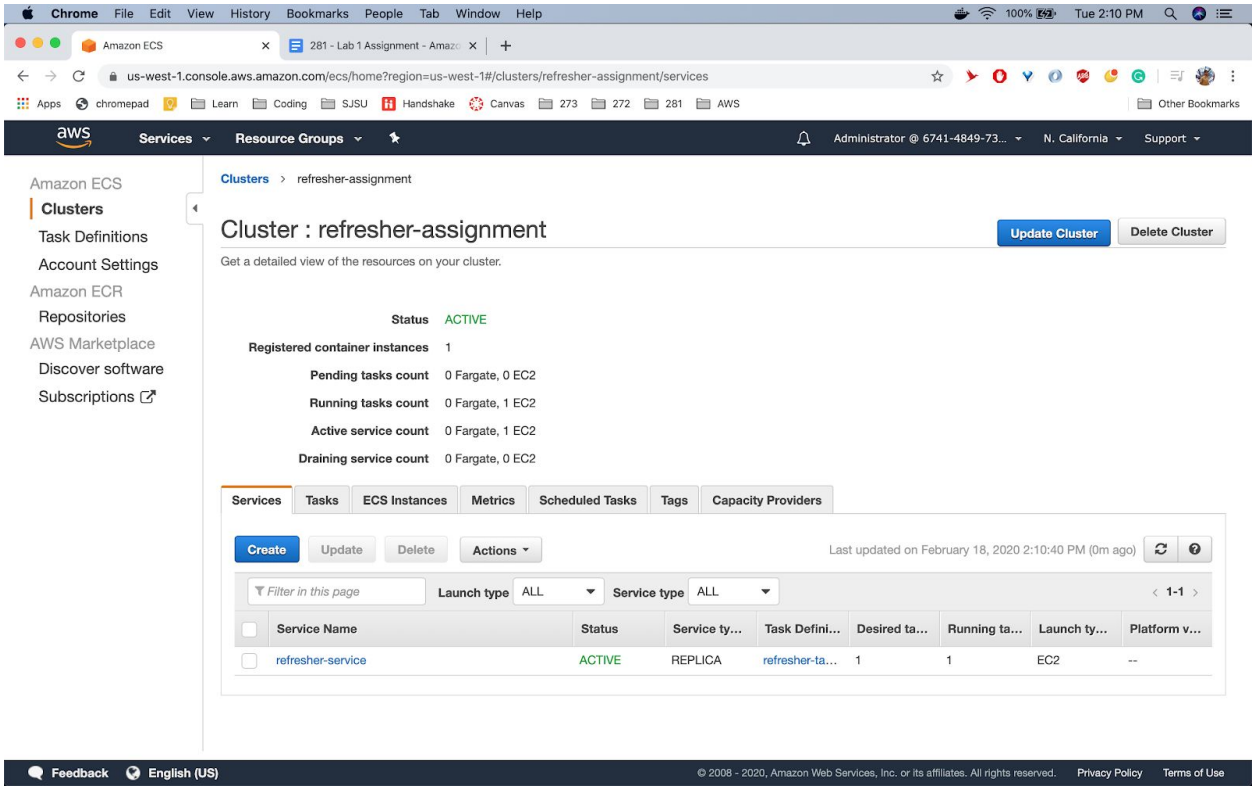
```
{
  "ipcMode": null,
  "executionRoleArn": null,
  "containerDefinitions": [
    {
      "dnsSearchDomains": null,
      "logConfiguration": null,
      "entryPoint": null,
      "portMappings": [
        {
          "hostPort": 3000,
          "protocol": "tcp",
          "containerPort": 3000
        }
      ],
      "command": null,
```

```
    "linuxParameters": null,
    "cpu": 1024,
    "environment": [],
    "ulimits": null,
    "dnsServers": null,
    "mountPoints": [],
    "workingDirectory": null,
    "secrets": null,
    "dockerSecurityOptions": null,
    "memory": 500,
    "memoryReservation": null,
    "volumesFrom": [],
    "stopTimeout": null,
    "image": "674148497356.dkr.ecr.us-west-1.amazonaws.com/refresher_assignment:latest",
    "startTimeout": null,
    "firelensConfiguration": null,
    "dependsOn": null,
    "disableNetworking": null,
    "interactive": null,
    "healthCheck": null,
    "essential": true,
    "links": null,
    "hostname": null,
    "extraHosts": null,
    "pseudoTerminal": null,
    "user": null,
    "readonlyRootFilesystem": null,
    "dockerLabels": null,
    "systemControls": null,
    "privileged": null,
    "name": "refresher-container"
  }
],
"placementConstraints": [],
"memory": null,
"taskRoleArn": null,
"compatibilities": [
  "EC2"
],
"taskDefinitionArn": "arn:aws:ecs:us-west-1:674148497356:task-definition/refresher-task:1",
"family": "refresher-task",
"requiresAttributes": [
  {
```

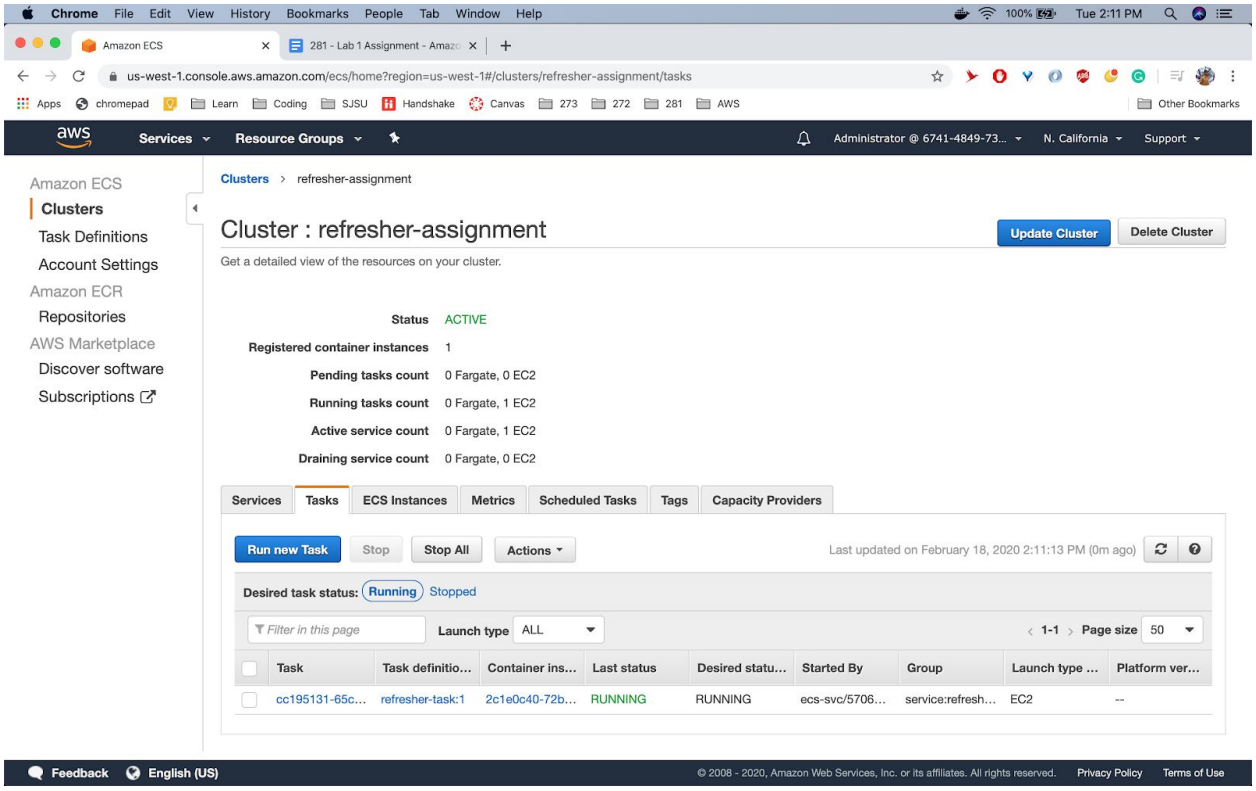
```
      "targetId": null,
      "targetType": null,
      "value": null,
      "name": "com.amazonaws.ecs.capability.ecr-auth"
    }
  ],
  "pidMode": null,
  "requiresCompatibilities": [
    "EC2"
  ],
  "networkMode": null,
  "cpu": null,
  "revision": 1,
  "status": "ACTIVE",
  "inferenceAccelerators": null,
  "proxyConfiguration": null,
  "volumes": []
}
```

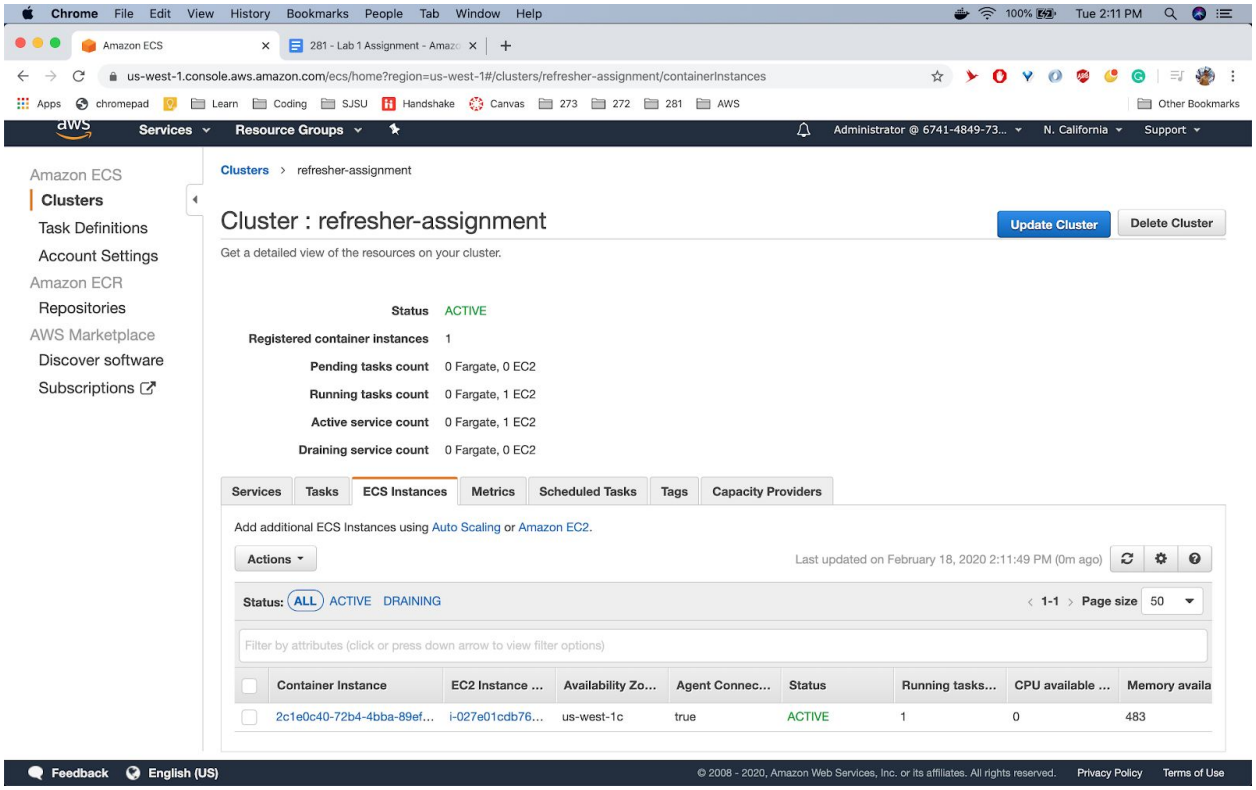

Next created a service for the task with below steps

1. Launch type - EC2
2. Assigned task definition
3. Assigned cluster
4. Entered service name

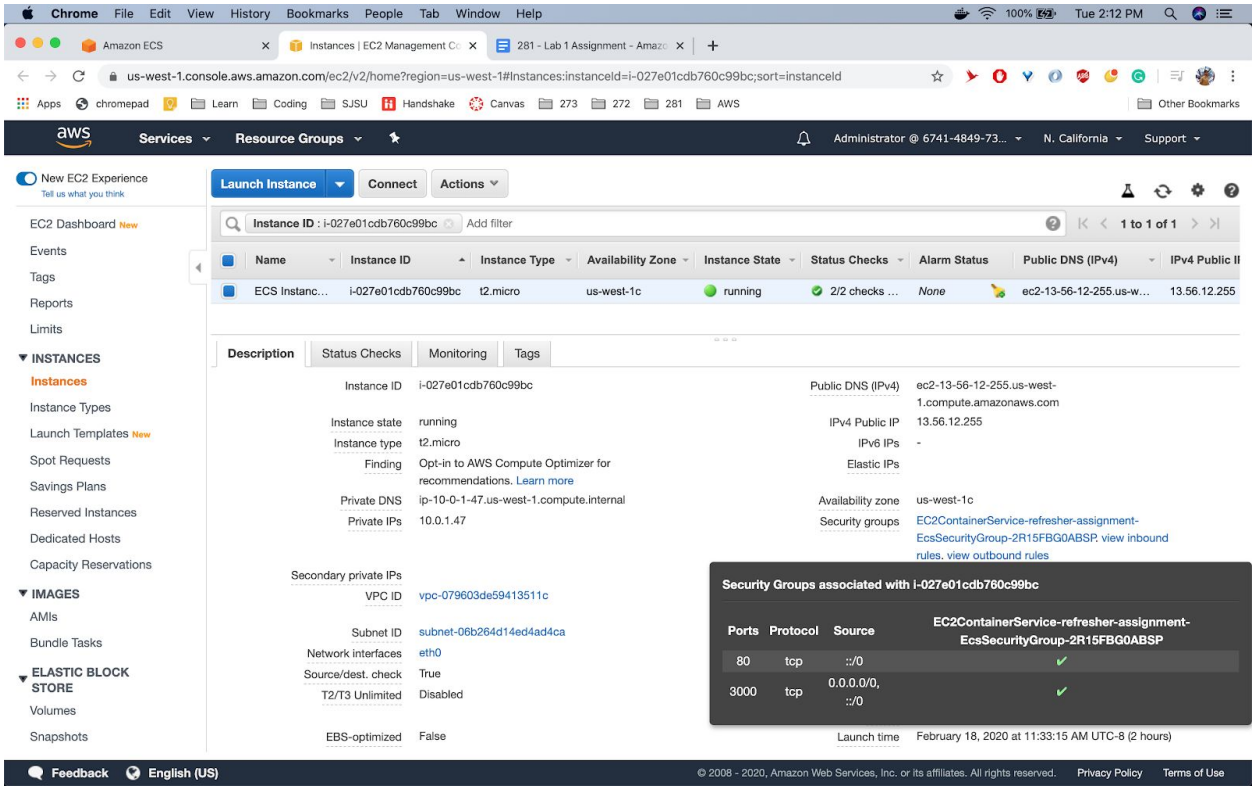


A task and an ecs instance got created

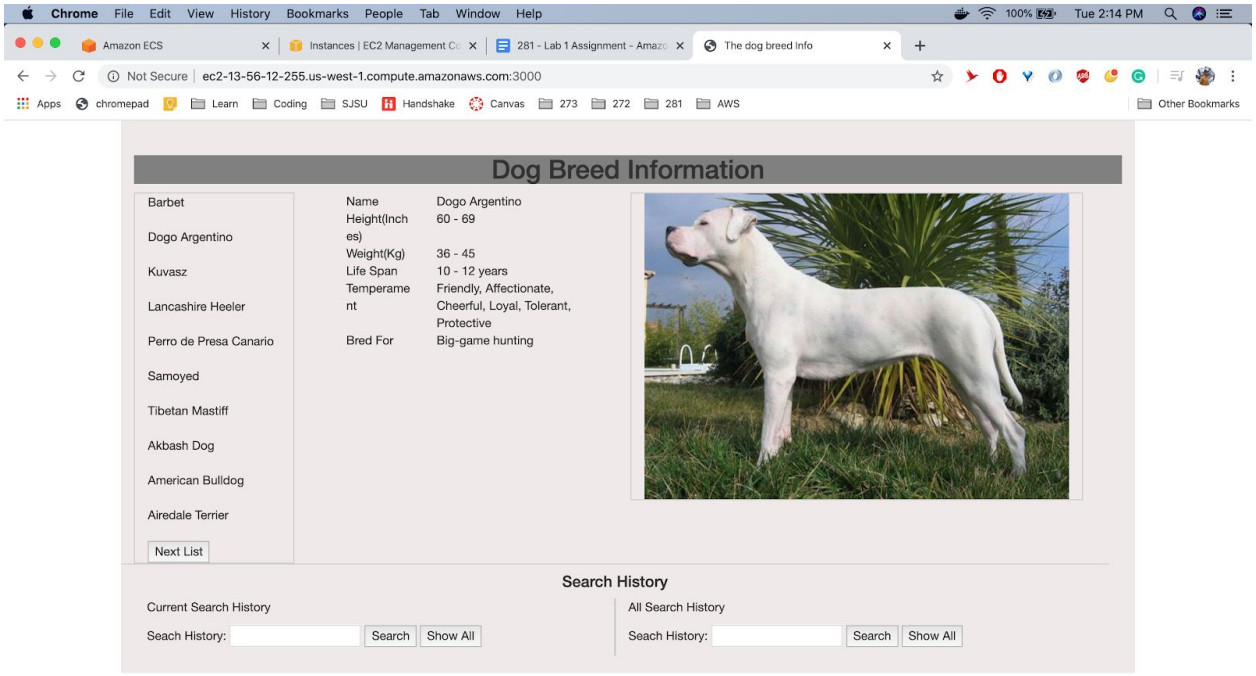




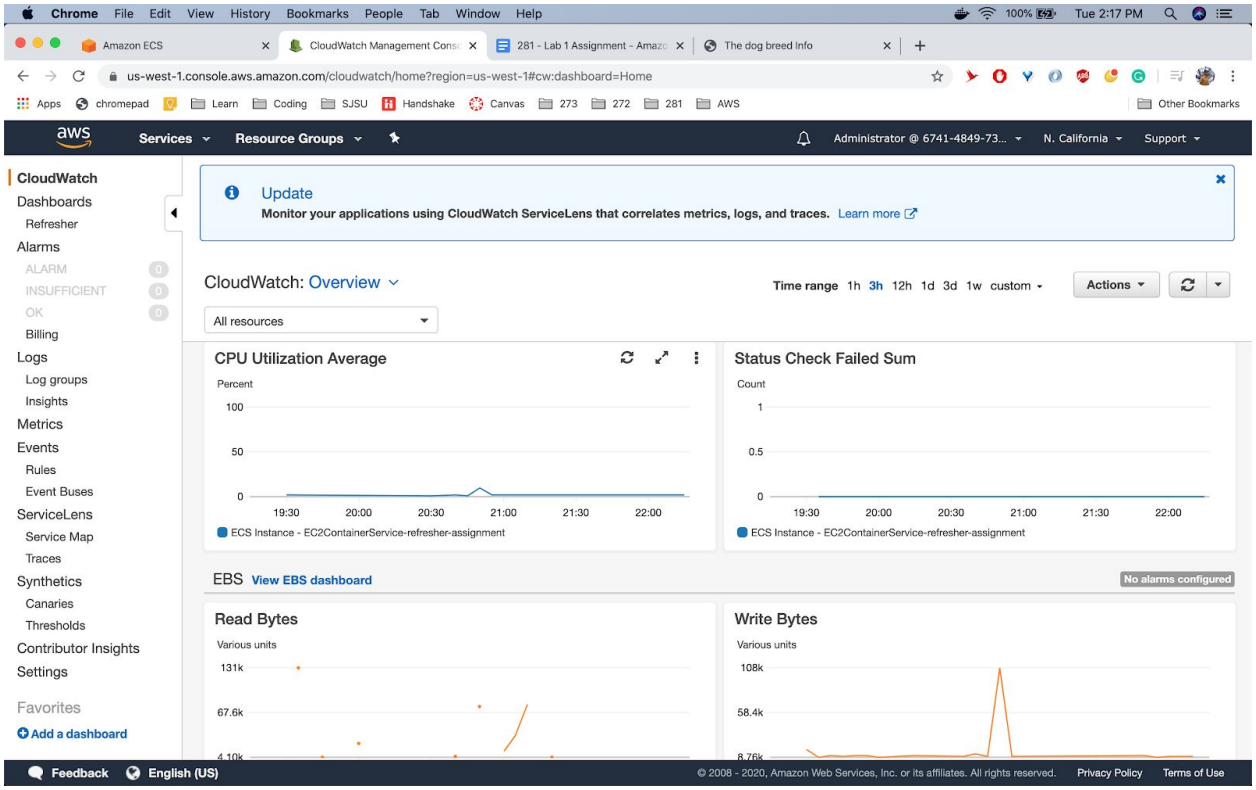
Added the inbound rule to EC2 instance to access the application



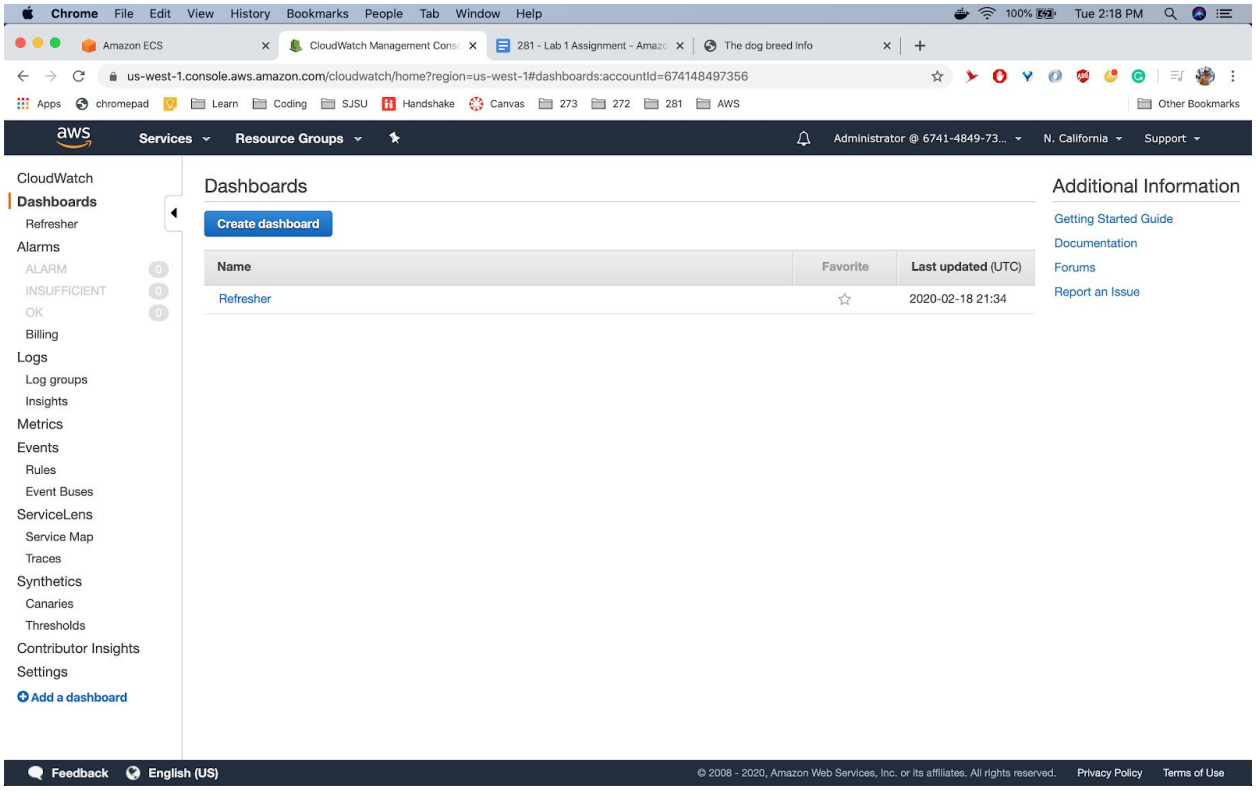
After that i was successfully able to launch the application in the browser

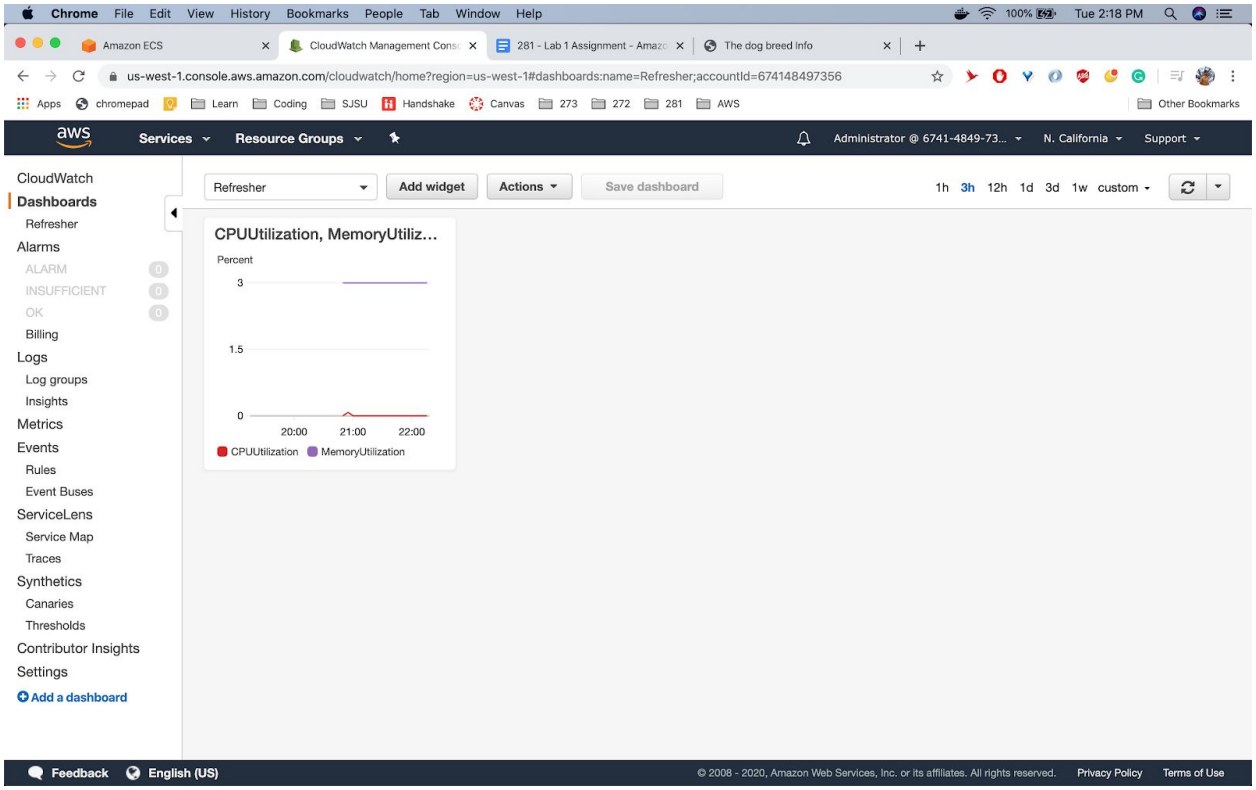


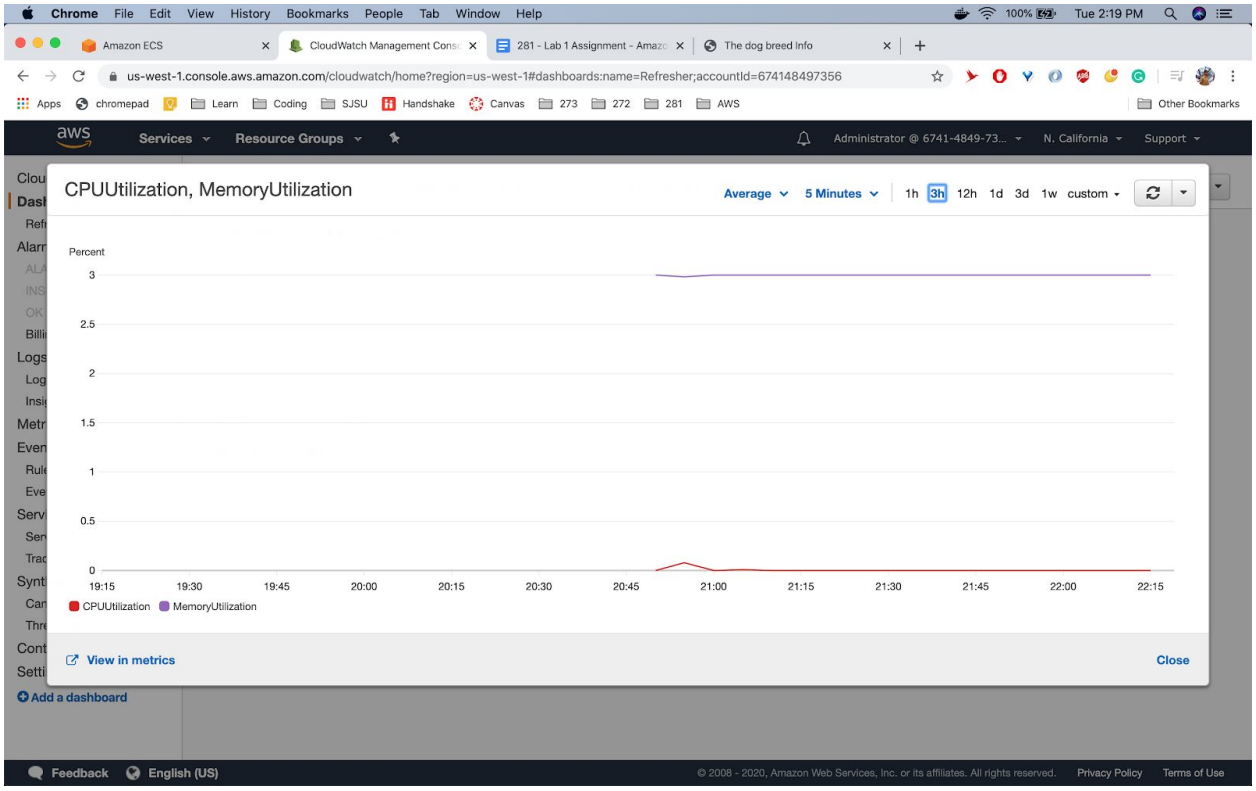
Using Cloudhwatch we can see how much the resources like CPU Memory and Storage are being used across all the instances



We can also customize to watch specific clusted by adding it to dashboard using Widget options




Chrome File Edit View History Bookmarks People Tab Window Help
Tue 2:18 PM
Amazon ECS
CloudWatch Management Cons
281 - Lab 1 Assignment - Amaz
The dog breed Info
us-west-1.console.aws.amazon.com/cloudwatch/home?region=us-west-1#dashboards:name=Refresher;accountId=674148497356
Apps chromepad Learn Coding SJSU Handshake Canvas 273 272 281 AWS
Other Bookmarks
aws
Services
Resource Groups
Administrator @ 6741-4849-73...
N. California
Support
CloudWatch
Dashboards
Refresher
Alarms
ALARM 0
INSUFFICIENT 0
OK 0
Billing
Logs
Log groups
Insights
Metrics
Events
Rules
Event Buses
ServiceLens
Service Map
Traces
Synthetics
Canaries
Thresholds
Contributor Insights
Settings
Add a dashboard
Refresher
Add widget
Actions
Save dashboard
1h 3h 12h 1d 3d 1w custom
CPUUtilization, MemoryUtiliz...
Percent
3
1.5
0
20:00
21:00
22:00
CPUUtilization
MemoryUtilization
Feedback
English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


Chrome File Edit View History Bookmarks People Tab Window Help
Tue 2:19 PM
us-west-1.console.aws.amazon.com/cloudwatch/home?region=us-west-1#dashboards:name=Refresher;accountId=674148497356
CPUUtilization, MemoryUtilization
Average
5 Minutes
1h 3h 12h 1d 3d 1w custom
Percent
3
2.5
2
1.5
1
0.5
0
19:15
19:30
19:45
20:00
20:15
20:30
20:45
21:00
21:15
21:30
21:45
22:00
22:15
CPUUtilization
MemoryUtilization
View in metrics
Close
Add a dashboard
Feedback
English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Since it is a very light weight application not much resources are being utilized the mejority of the memory is consumed due to the docker image deployed in the ec2 instance.