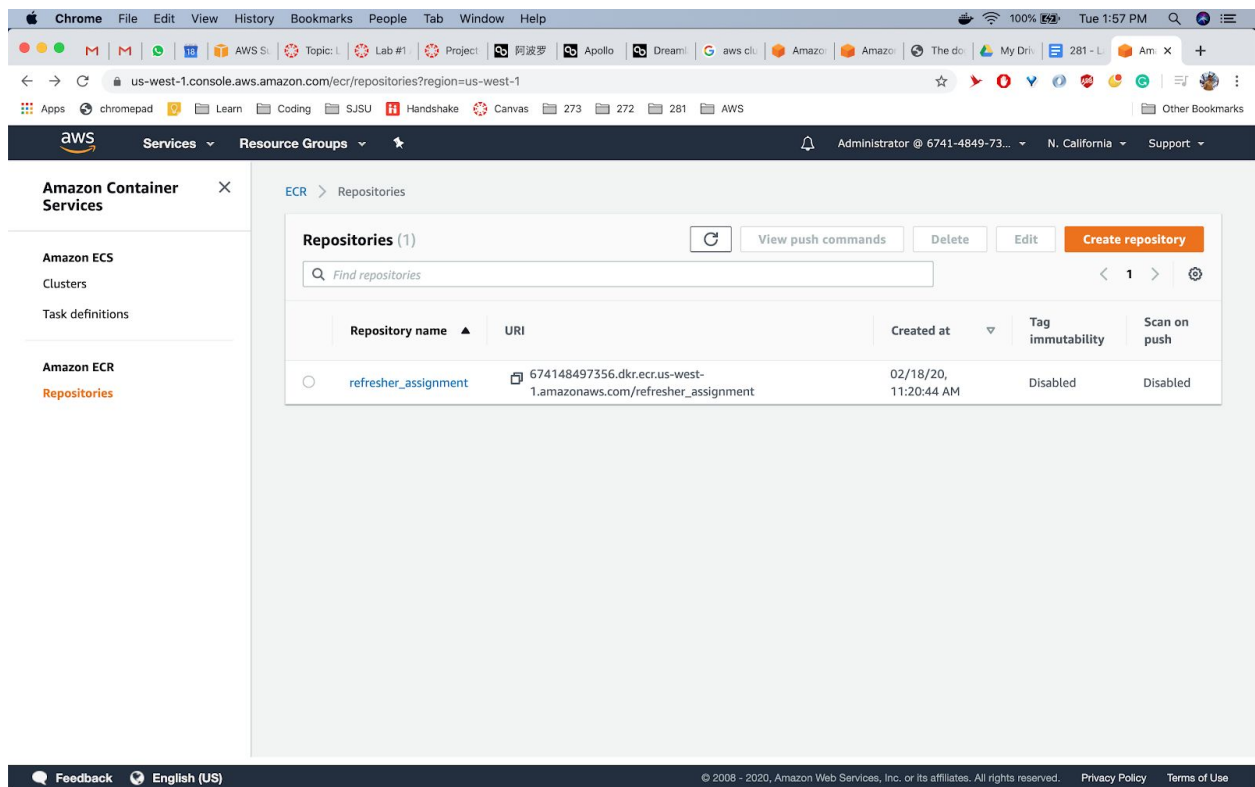


## Lab #1 Assignment - Amazon EC2

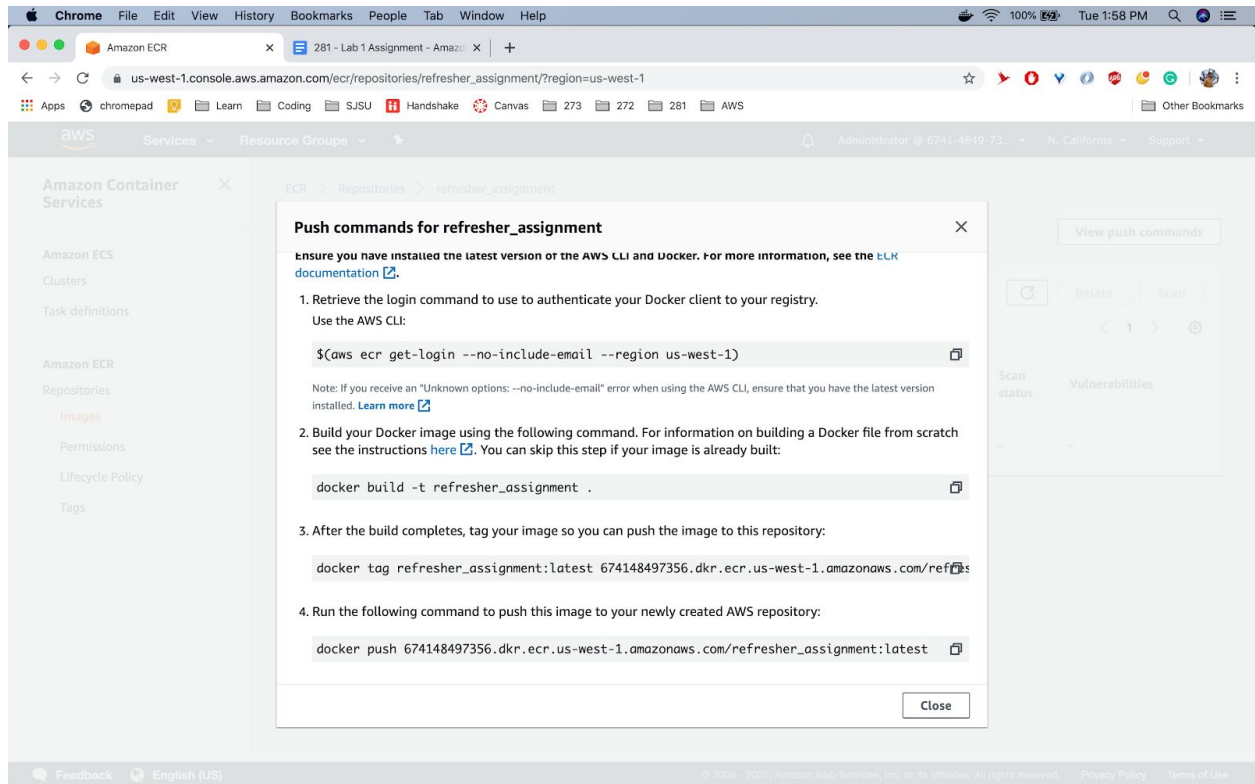
Here I will be describing how I created a simple web application and deployed it in AWS EC2 using docker and ECS.

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

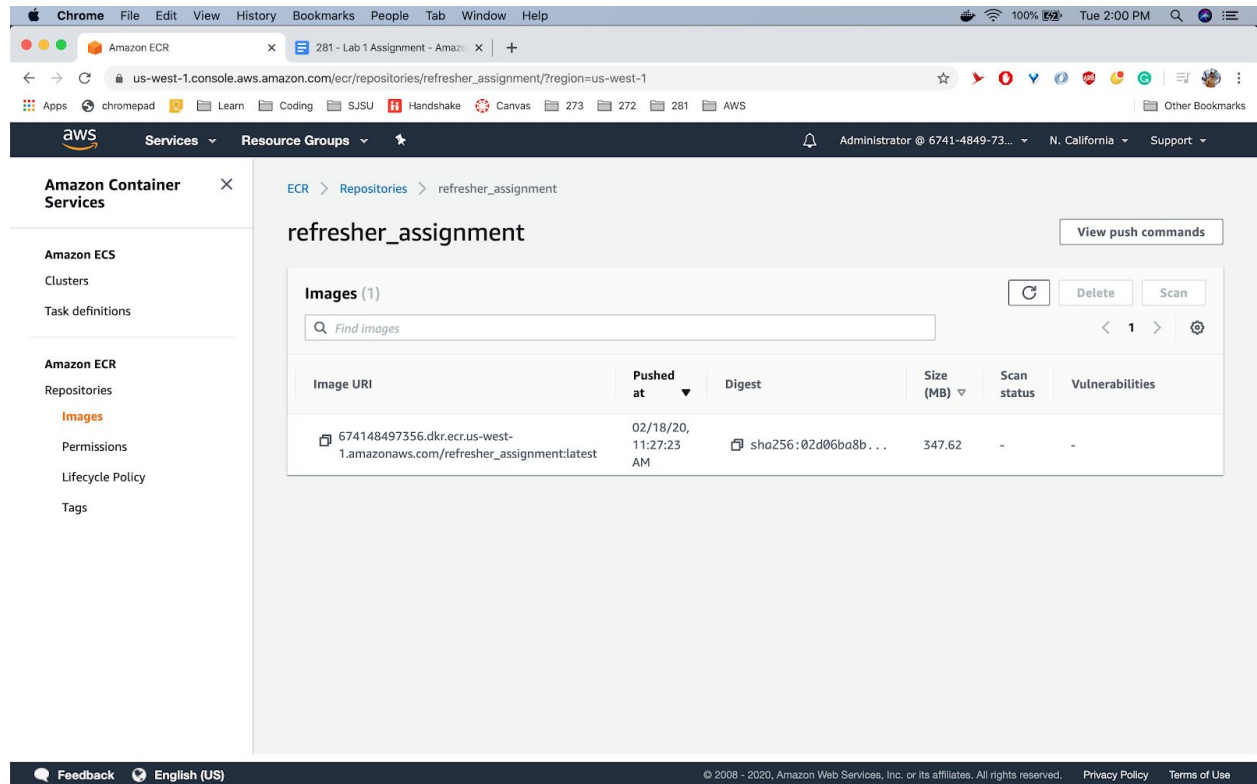


Which created an image and pushed it to the ECR

```
prashanth's MacBook-Air:app prashanthn$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
prashanth's 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>	da98cfffcd684	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		

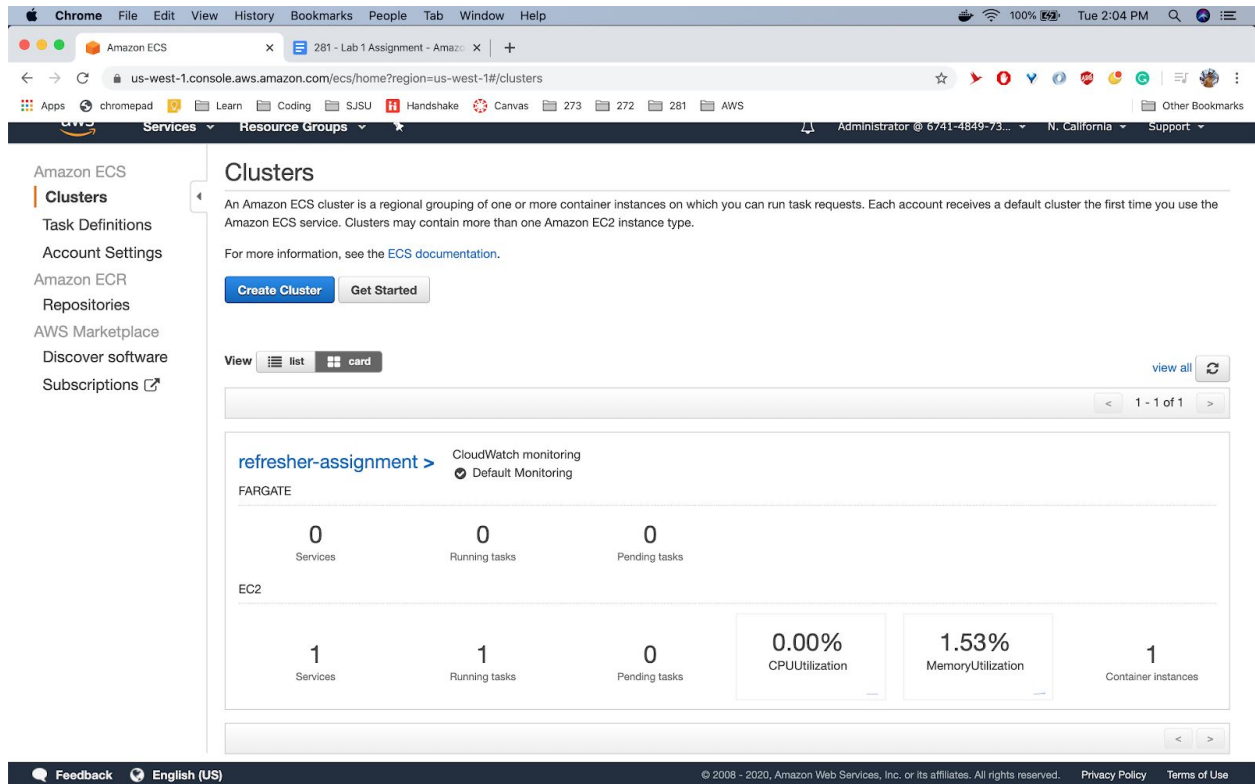
```
prashanth's MacBook-Air:app prashanthn$
```



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

Chrome File Edit View History Bookmarks People Tab Window Help

Amazon ECS 281 - Lab 1 Assignment - Amazon ECS

us-west-1.console.aws.amazon.com/ecs/home?region=us-west-1#/clusters/refresh-assignment/services

Services Resource Groups

Amazon ECS

- Clusters
- Task Definitions
- Account Settings

Amazon ECR

- Repositories
- AWS Marketplace
- Discover software
- Subscriptions

Clusters > refresh-assignment

## Cluster : refresh-assignment

Update Cluster Delete Cluster

Get a detailed view of the resources on your cluster.

Status **ACTIVE**

Registered container instances 1

Pending tasks count 0 Fargate, 0 EC2

Running tasks count 0 Fargate, 1 EC2

Active service count 0 Fargate, 1 EC2

Draining service count 0 Fargate, 0 EC2

Services Tasks ECS Instances Metrics Scheduled Tasks Tags Capacity Providers

Create Update Delete Actions

Last updated on February 18, 2020 2:10:40 PM (0m ago)

Filter in this page Launch type ALL Service type ALL

	Service Name	Status	Service ty...	Task Defini...	Desired ta...	Running ta...	Launch ty...	Platform v...
<input type="checkbox"/>	refresh-service	ACTIVE	REPLICA	refresh-ta...	1	1	EC2	--

Feedback English (US)

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

A task and an ecs instance got created

Chrome File Edit View History Bookmarks People Tab Window Help

Amazon ECS 281 - Lab 1 Assignment - Amazon ECS

us-west-1.console.aws.amazon.com/ecs/home?region=us-west-1#/clusters/refresh-assignment/tasks

Services Resource Groups

Amazon ECS

- Clusters
- Task Definitions
- Account Settings

Amazon ECR

- Repositories
- AWS Marketplace
- Discover software
- Subscriptions

Clusters > refresh-assignment

## Cluster : refresh-assignment

Update Cluster Delete Cluster

Get a detailed view of the resources on your cluster.

Status **ACTIVE**

Registered container instances 1

Pending tasks count 0 Fargate, 0 EC2

Running tasks count 0 Fargate, 1 EC2

Active service count 0 Fargate, 1 EC2

Draining service count 0 Fargate, 0 EC2

Services Tasks ECS Instances Metrics Scheduled Tasks Tags Capacity Providers

Run new Task Stop Stop All Actions

Last updated on February 18, 2020 2:11:13 PM (0m ago)

Desired task status: Running Stopped

Filter in this page Launch type ALL

	Task	Task definitio...	Container ins...	Last status	Desired statu...	Started By	Group	Launch type ...	Platform ver...
<input type="checkbox"/>	cc195131-65c...	refresh-task:1	2c1e0c40-72b...	RUNNING	RUNNING	ecs-svc/5706...	service:refresh...	EC2	--

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

Amazon ECS

us-west-1.console.aws.amazon.com/ecs/home?region=us-west-1#/clusters/refresher-assignment/containerInstances

Services Resource Groups

Administrator @ 6741-4849-73... N. California Support

Amazon ECS

Clusters

Task Definitions

Account Settings

Amazon ECR

Repositories

AWS Marketplace

Discover software

Subscriptions

Clusters > refresher-assignment

## Cluster : refresher-assignment

Get a detailed view of the resources on your cluster.

Status **ACTIVE**

Registered container instances 1

Pending tasks count 0 Fargate, 0 EC2

Running tasks count 0 Fargate, 1 EC2

Active service count 0 Fargate, 1 EC2

Draining service count 0 Fargate, 0 EC2

Services Tasks **ECS Instances** Metrics Scheduled Tasks Tags Capacity Providers

Add additional ECS Instances using [Auto Scaling](#) or [Amazon EC2](#).

Actions Last updated on February 18, 2020 2:11:49 PM (0m ago)

Status: **ALL** ACTIVE DRAINING

Filter by attributes (click or press down arrow to view filter options)

Container Instance	EC2 Instance ...	Availability Zo...	Agent Connec...	Status	Running tasks...	CPU available ...	Memory availa
<input type="checkbox"/> 2c1e0c40-72b4-4bba-89ef...	i-027e01cdb76...	us-west-1c	true	ACTIVE	1	0	483

Feedback English (US)

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Added the inbound rule to EC2 instance to access the application

Chrome File Edit View History Bookmarks People Tab Window Help

Amazon ECS

Instances | EC2 Management Co

281 - Lab 1 Assignment - Amaz

us-west-1.console.aws.amazon.com/ec2/v2/home?region=us-west-1#instances:instanceId=i-027e01cdb760c99bc;sort=instanceId

Services Resource Groups

Administrator @ 6741-4849-73... N. California Support

New EC2 Experience

Launch Instance Connect Actions

EC2 Dashboard

Events

Tags

Reports

Limits

INSTANCES

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

IMAGES

AMIs

Bundle Tasks

ELASTIC BLOCK STORE

Volumes

Snapshots

Instance ID: i-027e01cdb760c99bc

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public I
ECS Instanc...	i-027e01cdb760c99bc	t2.micro	us-west-1c	running	2/2 checks ...	None	ec2-13-56-12-255.us-w...	13.56.12.255

Description Status Checks Monitoring Tags

Instance ID i-027e01cdb760c99bc

Instance state running

Instance type t2.micro

Finding Opt-in to AWS Compute Optimizer for recommendations. [Learn more](#)

Private DNS ip-10-0-1-47.us-west-1.compute.internal

Private IPs 10.0.1.47

Secondary private IPs

VPC ID vpc-079603de59413511c

Subnet ID subnet-06b264d14ed4ad4ca

Network interfaces eth0

Source/dest. check True

T2/T3 Unlimited Disabled

EBS-optimized False

Public DNS (IPv4) ec2-13-56-12-255.us-west-1.compute.amazonaws.com

IPv4 Public IP 13.56.12.255

IPv6 IPs -

Elastic IPs

Availability zone us-west-1c

Security groups EC2ContainerService-refresher-assignment-EcsSecurityGroup-2R15FBG0ABSP. [view inbound rules.](#) [view outbound rules](#)

### Security Groups associated with i-027e01cdb760c99bc

Ports	Protocol	Source	EC2ContainerService-refresher-assignment-EcsSecurityGroup-2R15FBG0ABSP
80	tcp	:::0	✓
3000	tcp	0.0.0.0/0, :::0	✓

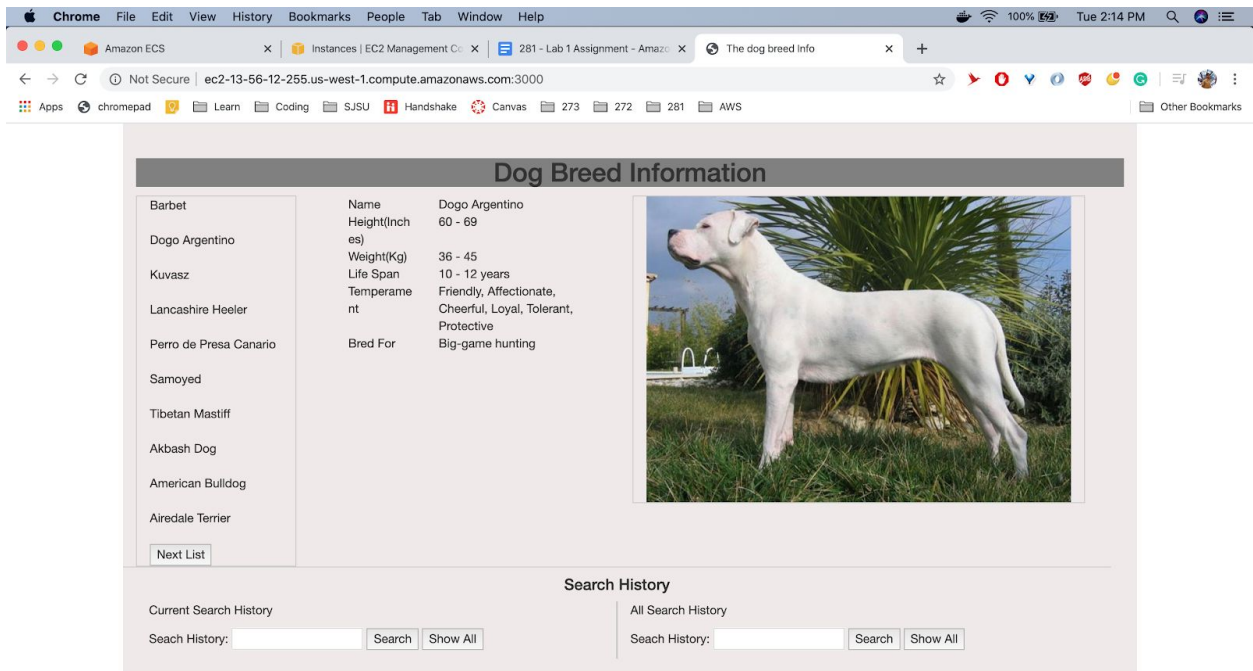
Launch time February 18, 2020 at 11:33:15 AM UTC-8 (2 hours)

Feedback English (US)

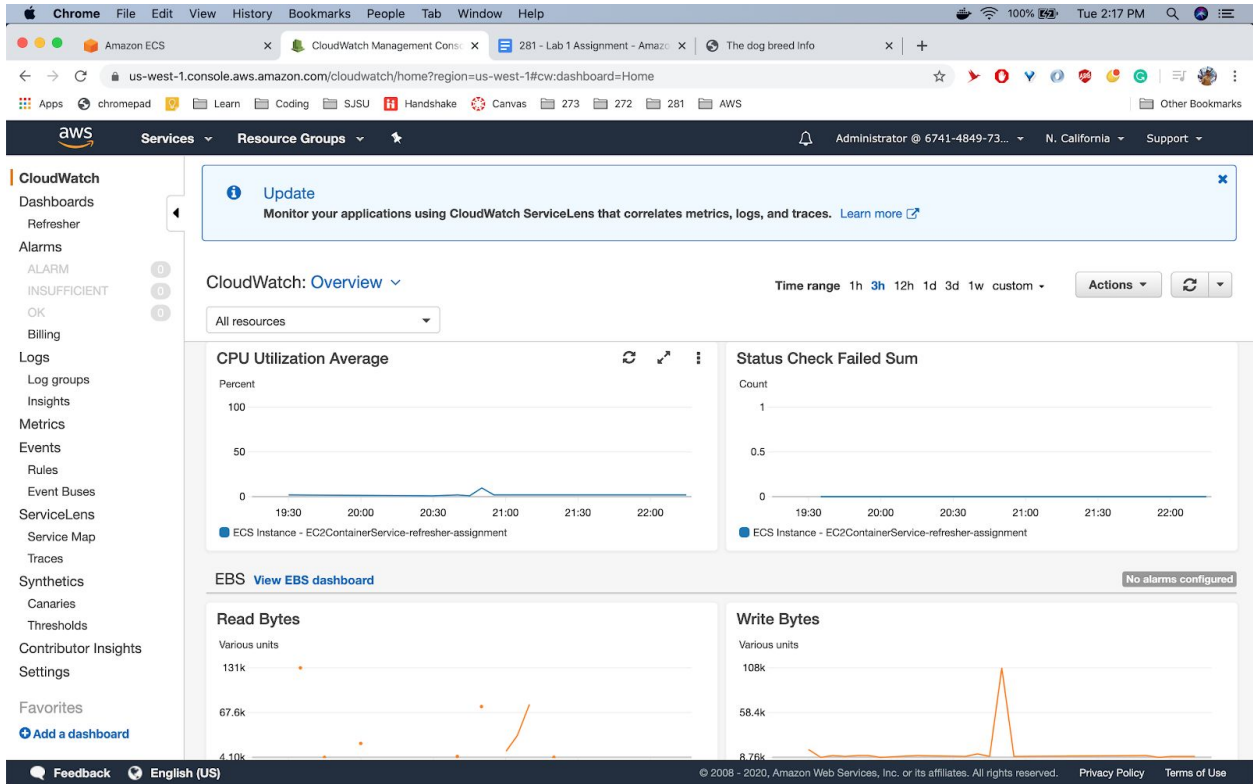
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use



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



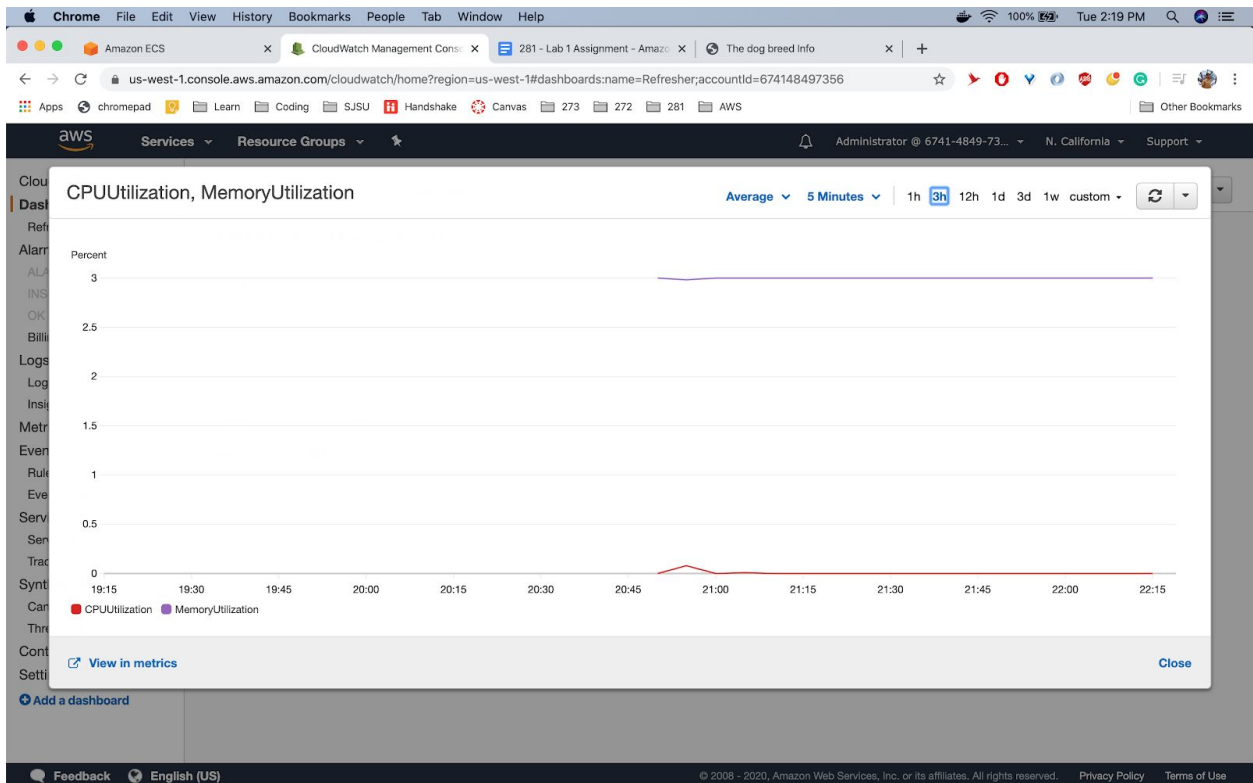
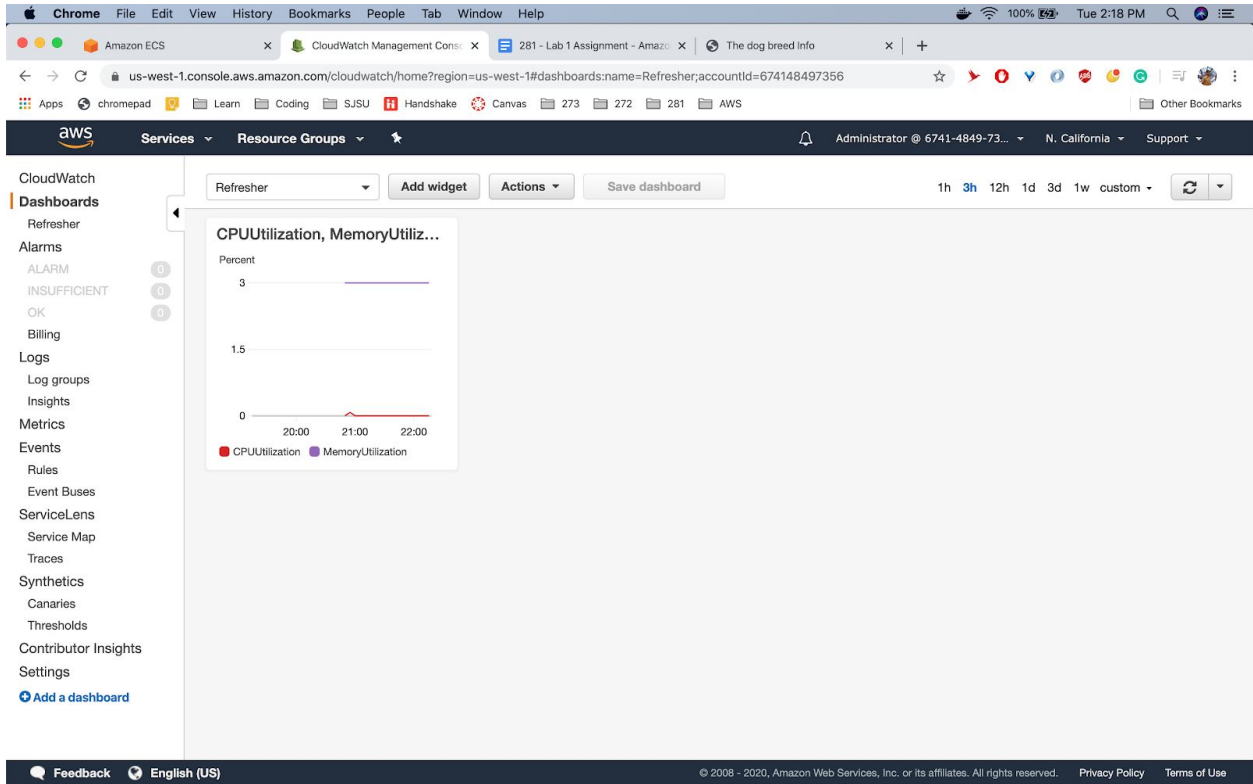
Using Cloudwatch we can see how much the resources like CPU Memory and Storage are being used across all the instances



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

The screenshot shows the AWS CloudWatch Dashboards page. It features a 'Create dashboard' button and a table listing existing dashboards. The table has columns for Name, Favorite, and Last updated (UTC). There is also an 'Additional Information' section on the right with links to guides, documentation, forums, and issue reporting.

Name	Favorite	Last updated (UTC)
Refresher	☆	2020-02-18 21:34



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