Docker - Trabajo práctico integrador

# Parte 1 - Deploy Local

Inicialmente se creó el volumen para la base de datos con el comando:

**docker volume create mse-db\_data**

Luego se creó una red con el comando:

**docker network create -d=bridge mse-network**

Luego con el comando

**docker run -d --name mongo --hostname mongo --network mse-network -v mse-db\_data:/data/db mongo:4.2.23**

Se crea y ejecuta un contenedor de Docker utilizando la imagen de MongoDB versión 4.2.23.

El contenedor se conectará a una red llamada "mse-network", se le asignará el nombre "mongo", se establecerá el nombre de host dentro del contenedor como "mongo" y se creará un volumen llamado "mse-db\_data" para almacenar los datos de la base de datos de MongoDB. El contenedor se ejecutará en segundo plano (modo "detach") y podrá ser accedido mediante el nombre "mongo".

Para verificar que este creado y en ejecución el contenedor se puede usar el comando

**docker ps**

Para construir la imagen mse-app, se modificó el Dockerfile, y se ejecuto el comando

**docker build -t mse-app .**

Posteriormente se inició la ejecución de la aplicación con el comando

**docker run -d --name mse-app \**

**--network mse-network \**

**--hostname mse-app \**

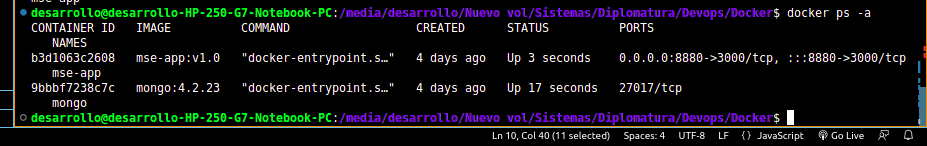
**-e MONGO\_HOST=mongo \**

**-p 8880:3000 \**

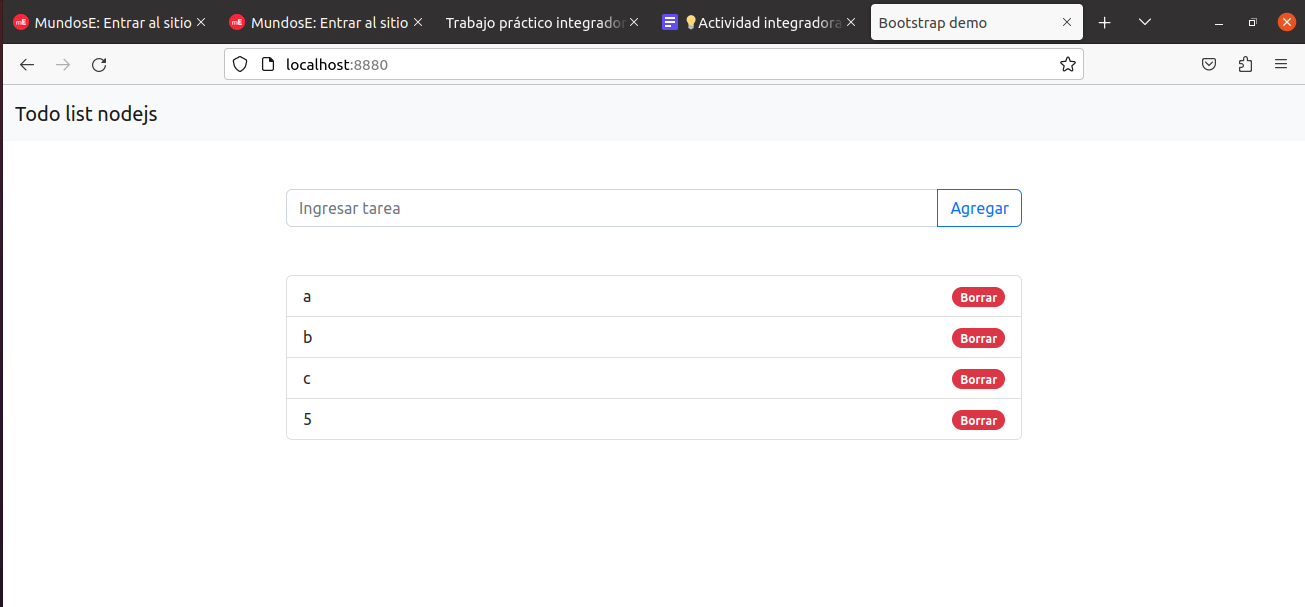
**mse-app:v1.0**

Nota: Se modificó el puerto 8080 por el 8880 por que ya esta en uso en el ordenador donde realizo mis actividades.

Para verificar que están creados los 2 contenedores se usa docker ps -a, los contenedores están en ejecución.

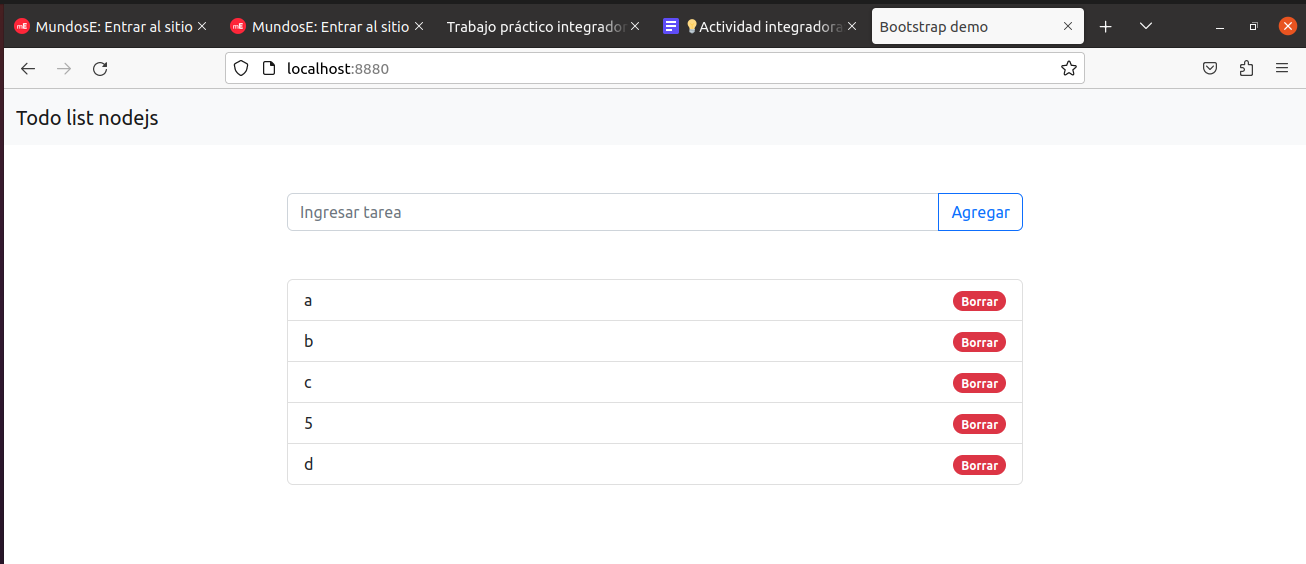


Luego se ingreso desde un navegador a la url <http://localhost:8880/>

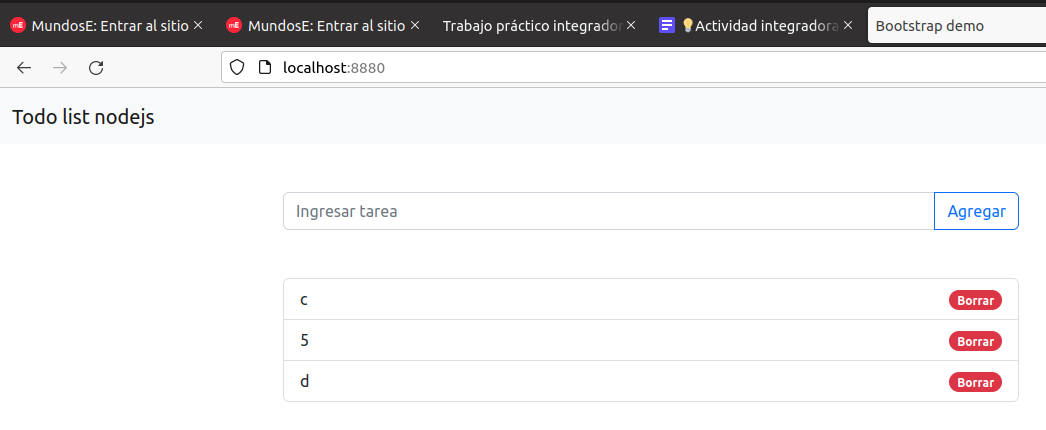


Donde se puede ver la aplicación y los datos ya creados, se creó un registro nuevo

con la letra d para verificar el funcionamiento.

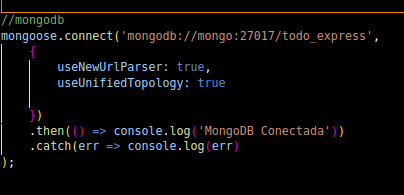


al igual que eliminar los registros “a” y “b”



# Desafios encontrados:

En el archivo de app.js se detecto que la coneccion que se esta estableciendo es con **mongo:27017** por lo que cuando se intento crear el contenedor de Mongo con   
**--name mse-db --hostname mse-db**, esto arrojaba error de conexión por lo que tome la decisión de modificar el nombre y el hostname del contenedor de Mongo a mongo como esta configurado en la app, de esta forma la aplicación si pudo conectarse con Mongo



Tambien se puede solucionar modificando la conexion de mongo ej

mongoose.connect('mongodb://mse-db:27017/todo\_express',

{

useNewUrlParser: true,

useUnifiedTopology: true

})

.then(() => console.log('MongoDB Conectada'))

.catch(err => console.log(err)

);

en ese caso se deben realizar las modificaciones en los comandos para usar mse-db en vez de mongo.

# Parte 2 - Deploy AWS

CONFIGURACIÓN AWS:

Se crearon las variables de entorno para realizar la tarea del práctico con los comandos.

**export AWS\_PROFILE=joel-deandres**

**export AWS\_REGION=us-west-2**

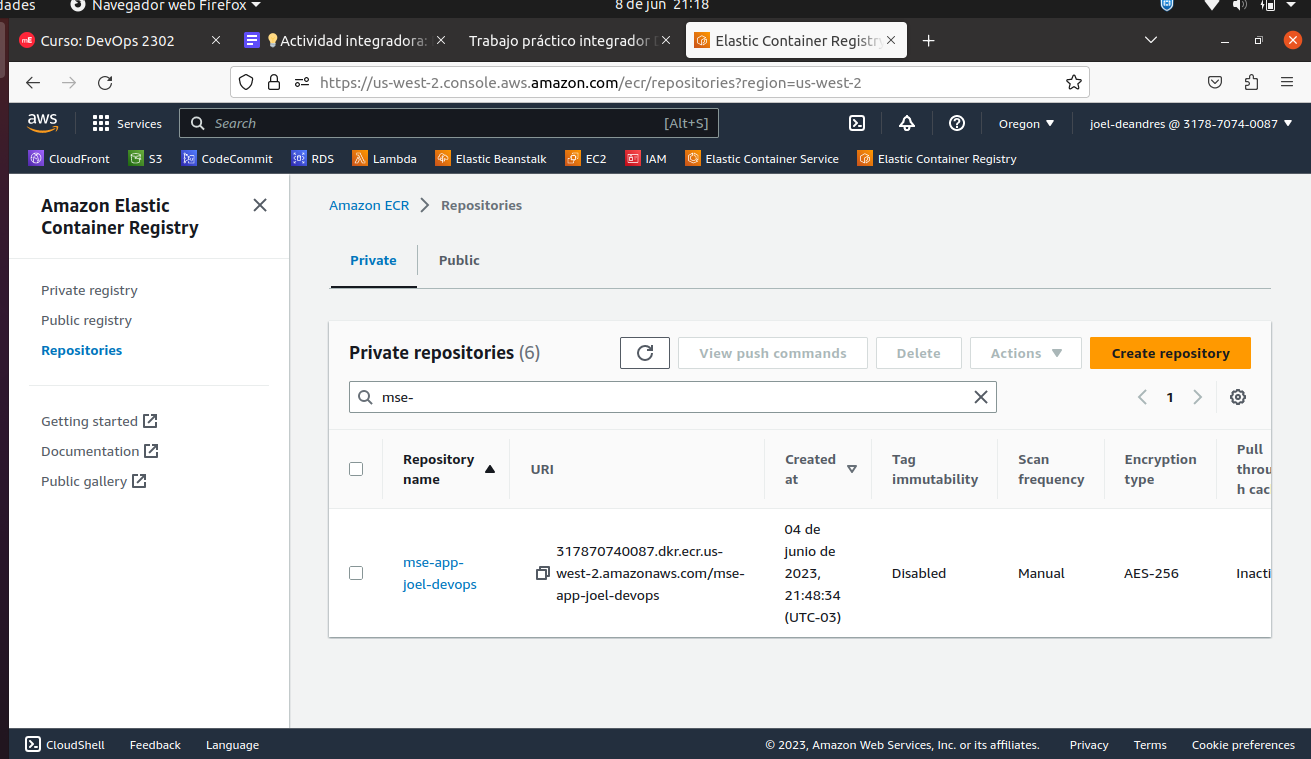
**export AWS\_ACCOUNT\_ID=317870740087**

En el servicio ECR se creó el registry **mse-app-joel-devops**

URI: ${AWS\_ACCOUNT\_ID}.dkr.ecr.us-west-2.amazonaws.com/mse-app-joel-devops

Para crear el registry se usó el comando

**aws ecr create-repository --repository-name mse-app-joel-devops**



En este registry se subieron 2 imágenes con los comandos:

**docker tag mse-app:v1.0 \**

**${AWS\_ACCOUNT\_ID}.dkr.ecr.${AWS\_REGION}.amazonaws.com/mse-app-joel-devops:v1.0**

**docker tag mse-app:v1.1 \**

**${AWS\_ACCOUNT\_ID}.dkr.ecr.${AWS\_REGION}.amazonaws.com/mse-app-joel-devops:v1.1**

Con estos comandos se etiquetaron las imagenes locales “mse-app:v1.0” y “mse-app:v1.1” con una nueva etiqueta “mse-app-joel-devops:v1.0” y “mse-app-joel-devops:v1.1” que corresponde a la ubicación en Amazon ECR.

**aws ecr get-login-password --region ${AWS\_REGION} | \**

**docker login --username AWS \**

**--password-stdin ${AWS\_ACCOUNT\_ID}.dkr.ecr.${AWS\_REGION}.amazonaws.com**

En este comando se obtiene el token de acceso al ECR utilizando AWS CLI y luego usar ese token para autenticarse en ECR utilizando Docker. Esto permite realizar operaciones relacionadas con imágenes de contenedor en ECR, como subir y descargar imágenes.

**docker push \**

**${AWS\_ACCOUNT\_ID}.dkr.ecr.${AWS\_REGION}.amazonaws.com/mse-app-joel-devops:v1.0**

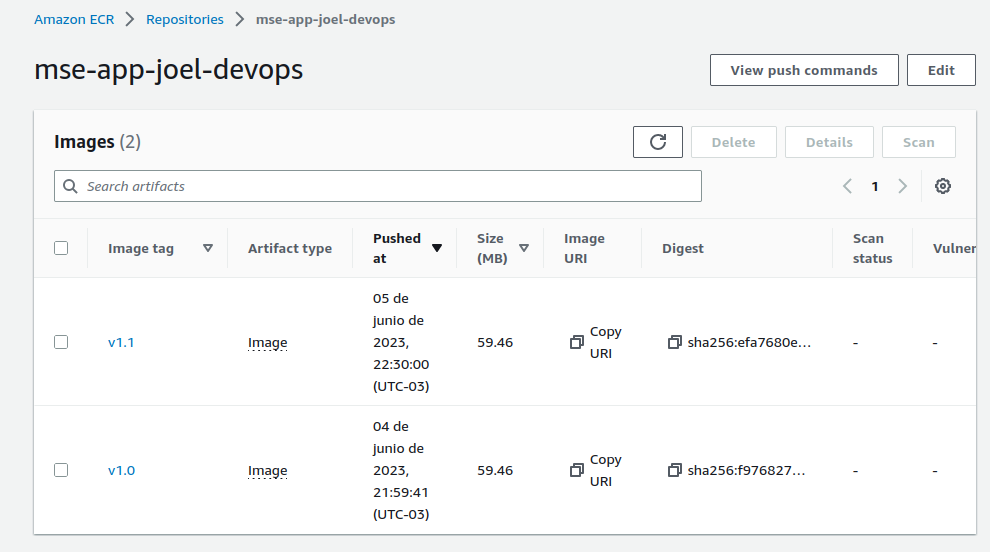
**docker push \**

**${AWS\_ACCOUNT\_ID}.dkr.ecr.${AWS\_REGION}.amazonaws.com/mse-app-joel-devops:v1.1**

En estos comandos se envían las imágenes de Docker al ECR AWS. La imagen se identifica por su nombre y etiqueta “mse-app-joel-devops:v1.0” y “mse-app-joel-devops:v1.1”, y se envía al registro para su almacenamiento y posterior uso en despliegues de contenedores.

**aws ecr describe-images --repository-name mse-app-joel-devops**

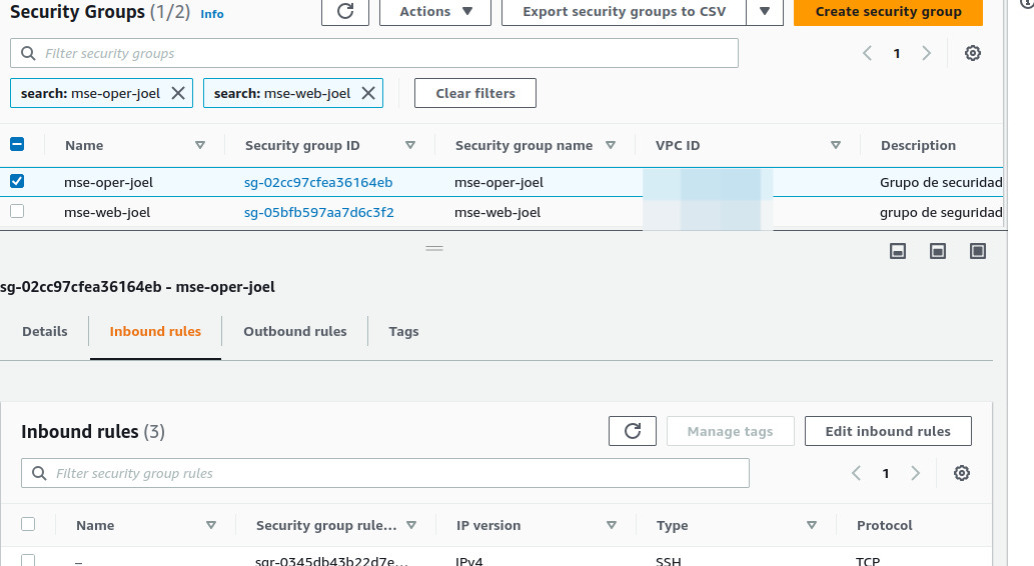
Este comando se utilizó para validar información sobre las imágenes almacenadas en ECR.



CREAR GRUPOS DE SEGURIDAD

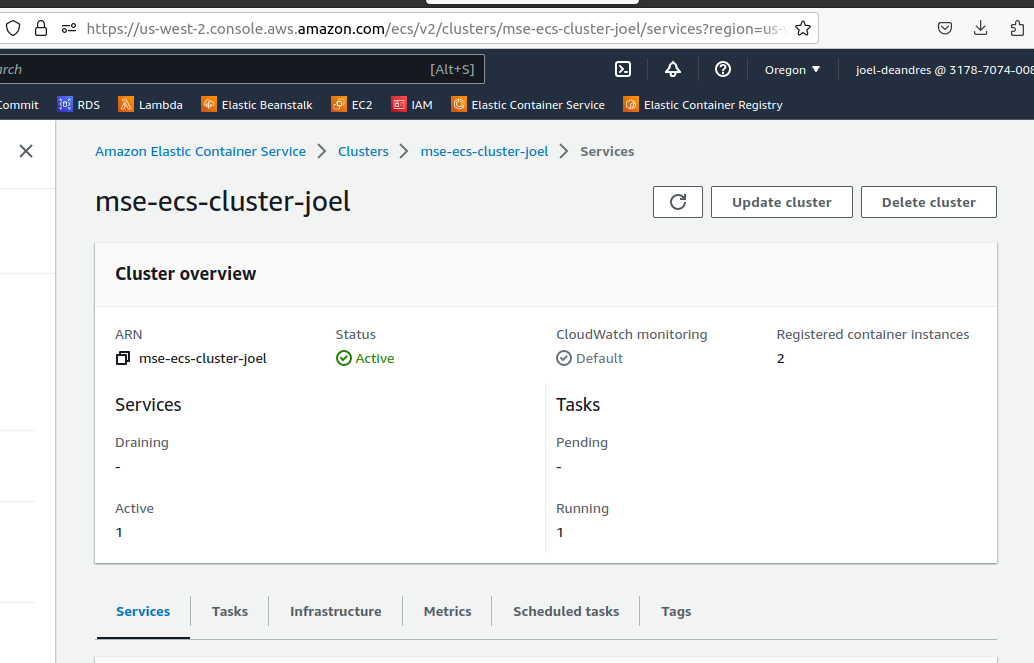
Dentro del servicio EC2 se crearon los grupos de seguridad

**mse-oper-joel**,**mse-web-joel** estos grupos de seguridad se configuraron para gestionar los permisos de acceso a los contenedores o instancias de AWS.

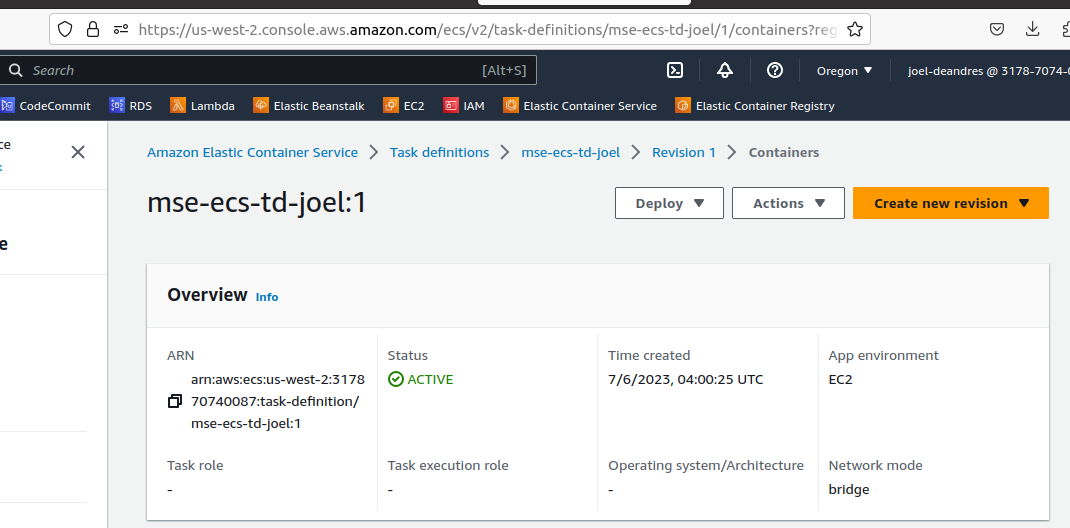


En el servicio de ECS se creó un CLUSTER con el nombre **mse-ecs-cluster-joel**

para configurar las tareas y servicios necesarios para montar los contenedores de la app.

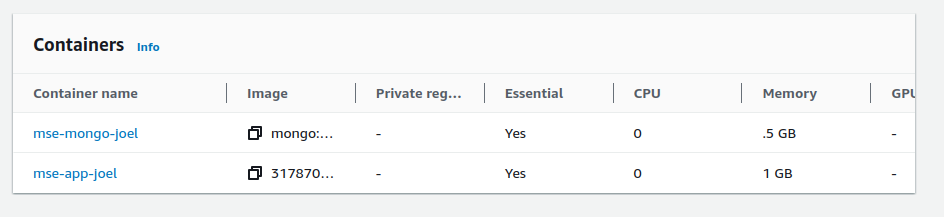


En TASK se creó **mse-ecs-td-joel** y el VOLUMEN **mse-db\_data**

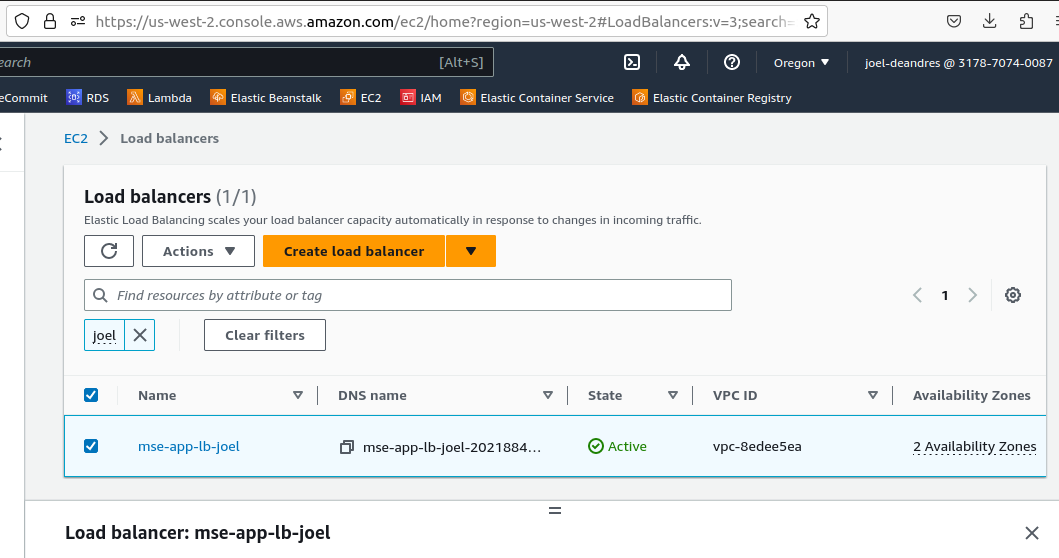
****

Se crearon Containers instances **mse-mongo-joel** y **mse-app-joel**

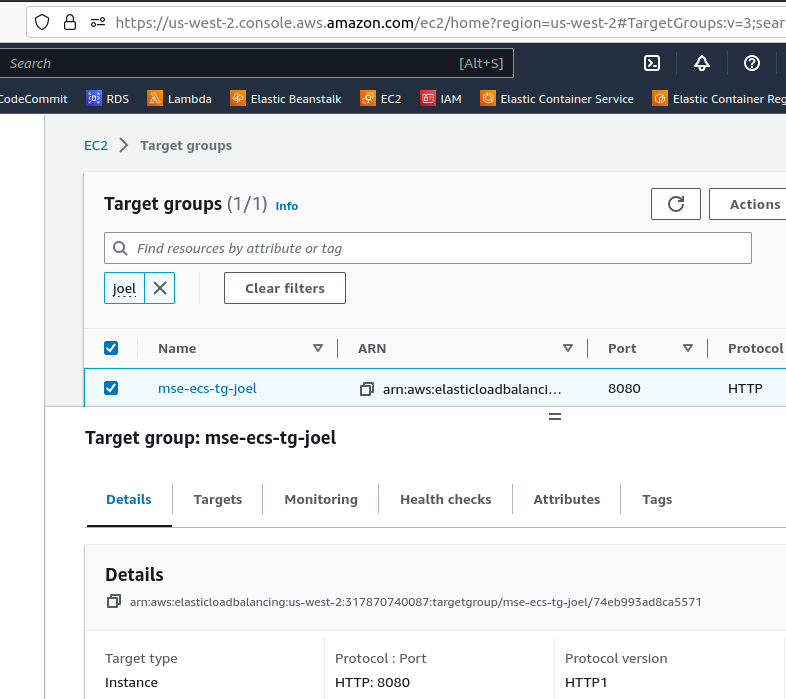
para instalar mongo:4.2.23 en **mse-mongo-joel** y la aplicación en **mse-app-joel**

****

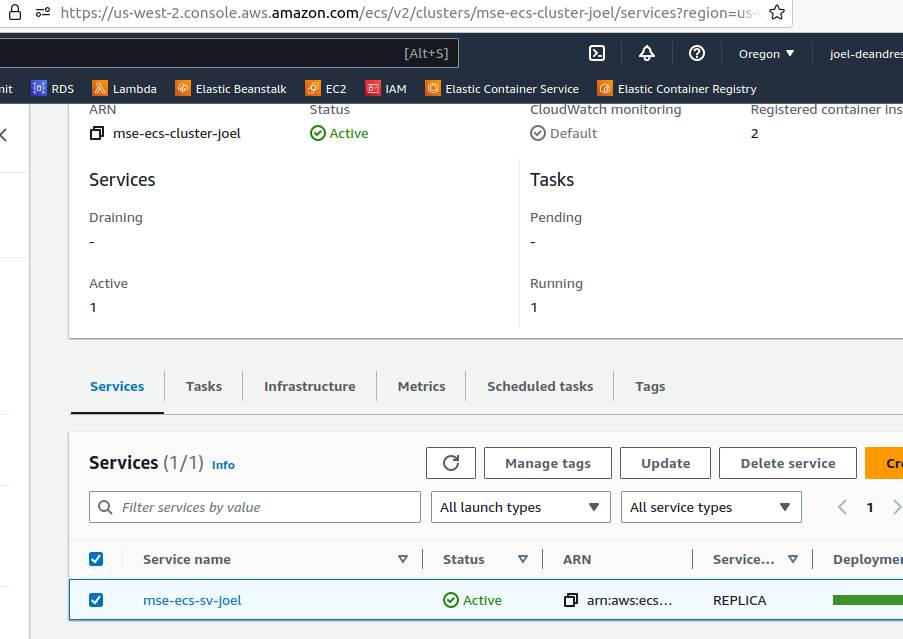
En el servicio de EC2 se creo un Load Balancer con el nombre **mse-app-lb-joel**

****

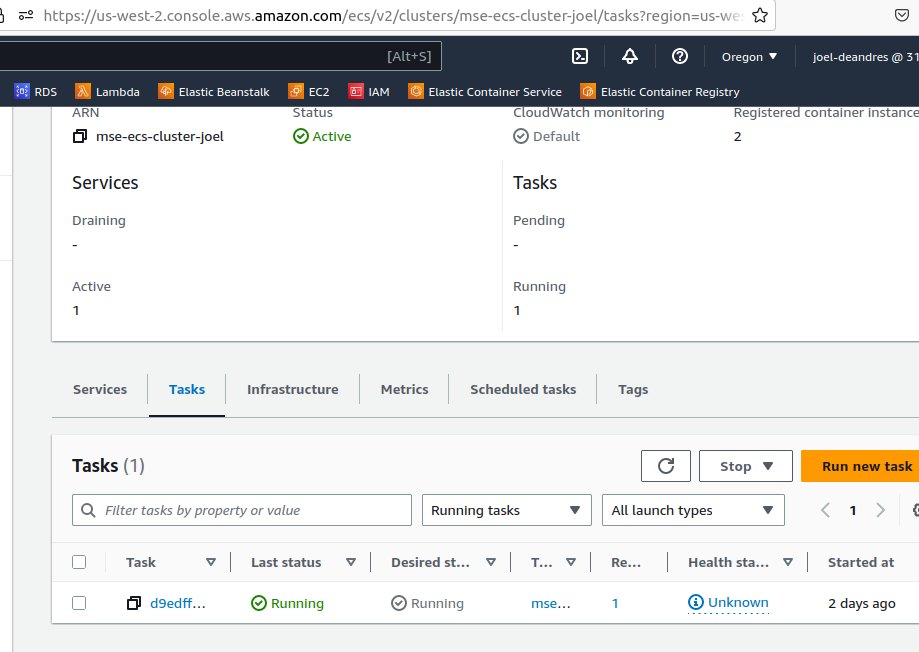
y un Target Group con el nombre **mse-ecs-tg-joel**

****

En el cluster **mse-ecs-cluster-joel** se creo el servicio con el nombre **mse-ecs-sv-joel**



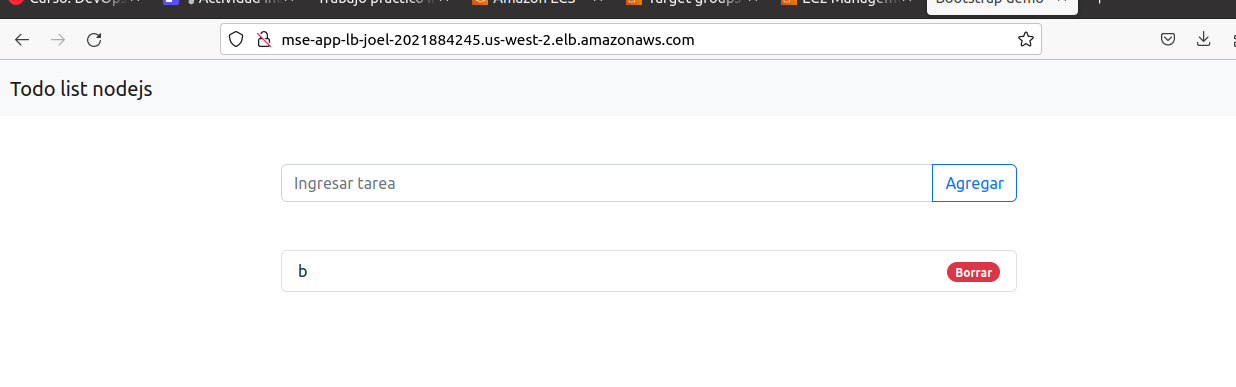
Se verificó que la tarea esté en ejecución



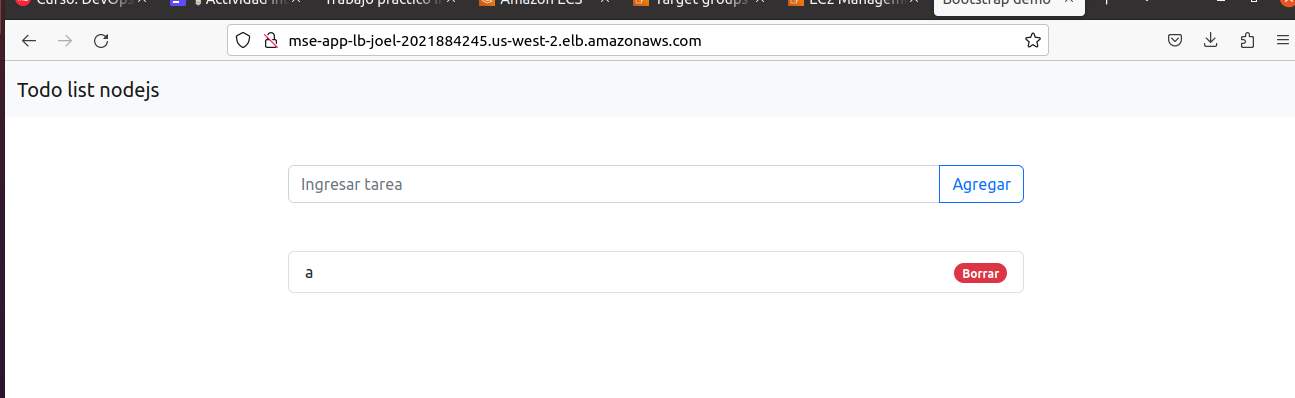
Para probar la aplicación se ingreso a la DNS que provee AWS en elb

URL:

http://mse-app-lb-joel-2021884245.us-west-2.elb.amazonaws.com/



En las pruebas se creó un registro “b” para verificar el funcionamiento se eliminó “b” y se creó “a”



NOTA:  
Por motivos de mantenimiento en la cuenta de aws este contenido creado para el práctico va a estar vigente por un periodo de tiempo y se eliminará.