

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.

```
desarrollo@desarrollo-HP-250-G7-Notebook-PC:/media/desarrollo/Nuevo vol/Sistemas/Diplomatura/Devops/Docker$ docker ps -a
```

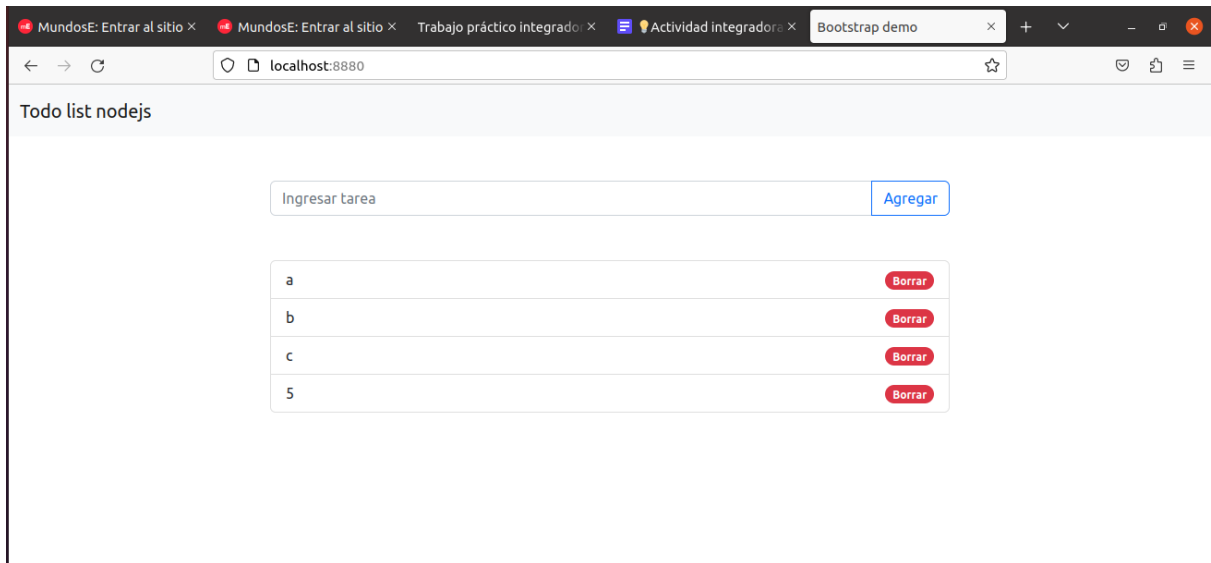
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
b3d1063c2608	mse-app:v1.0	"docker-entrypoint.s..."	4 days ago	Up 3 seconds	0.0.0.0:8880->3000/tcp, :::8880->3000/tcp
9bbbf7238c7c	mongo:4.2.23	"docker-entrypoint.s..."	4 days ago	Up 17 seconds	27017/tcp

```
desarrollo@desarrollo-HP-250-G7-Notebook-PC:/media/desarrollo/Nuevo vol/Sistemas/Diplomatura/Devops/Docker$
```

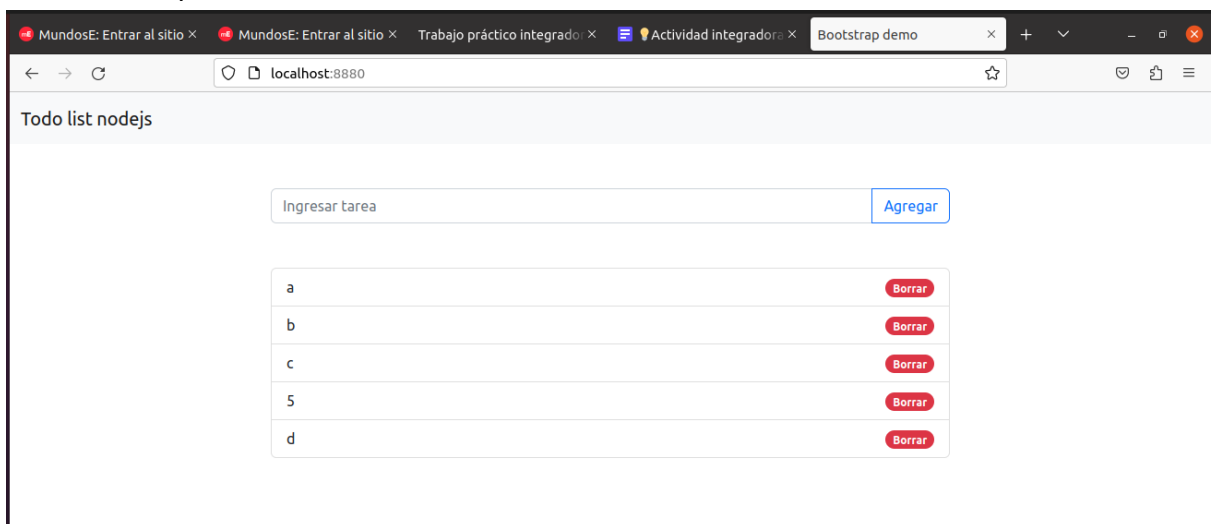
Ln 10, Col 40 (11 selected) Spaces: 4 UTF-8 LF {} JavaScript Go Live

Actividad integradora: Docker

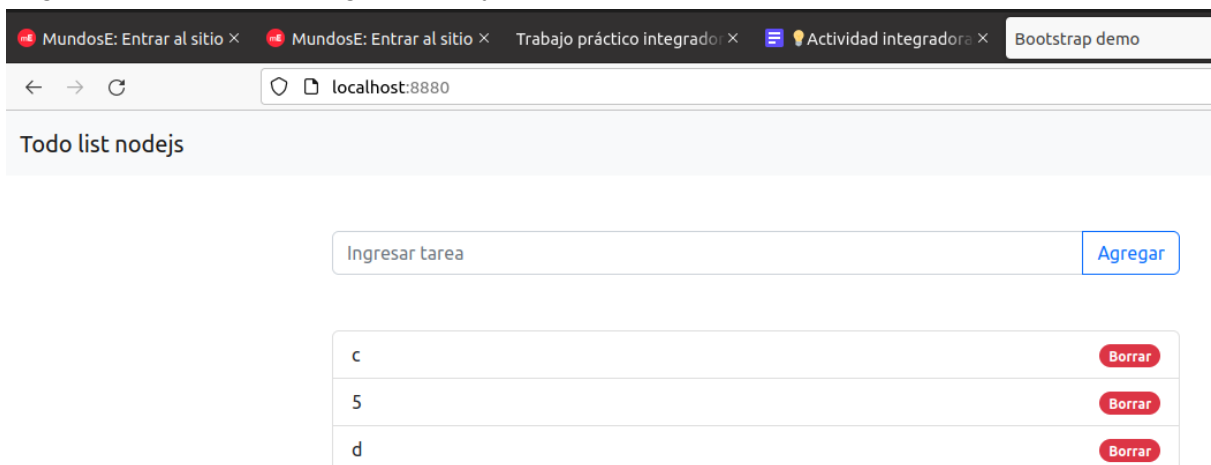
Luego se ingreso desde un navegador a la url <http://localhost:8880/>



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”





Actividad integradora: Docker

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

```
//mongodb
mongoose.connect('mongodb://mongo:27017/todo_express',
  {
    useNewUrlParser: true,
    useUnifiedTopology: true
  })
.then(() => console.log('MongoDB Conectada'))
.catch(err => console.log(err))
);
```

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
```

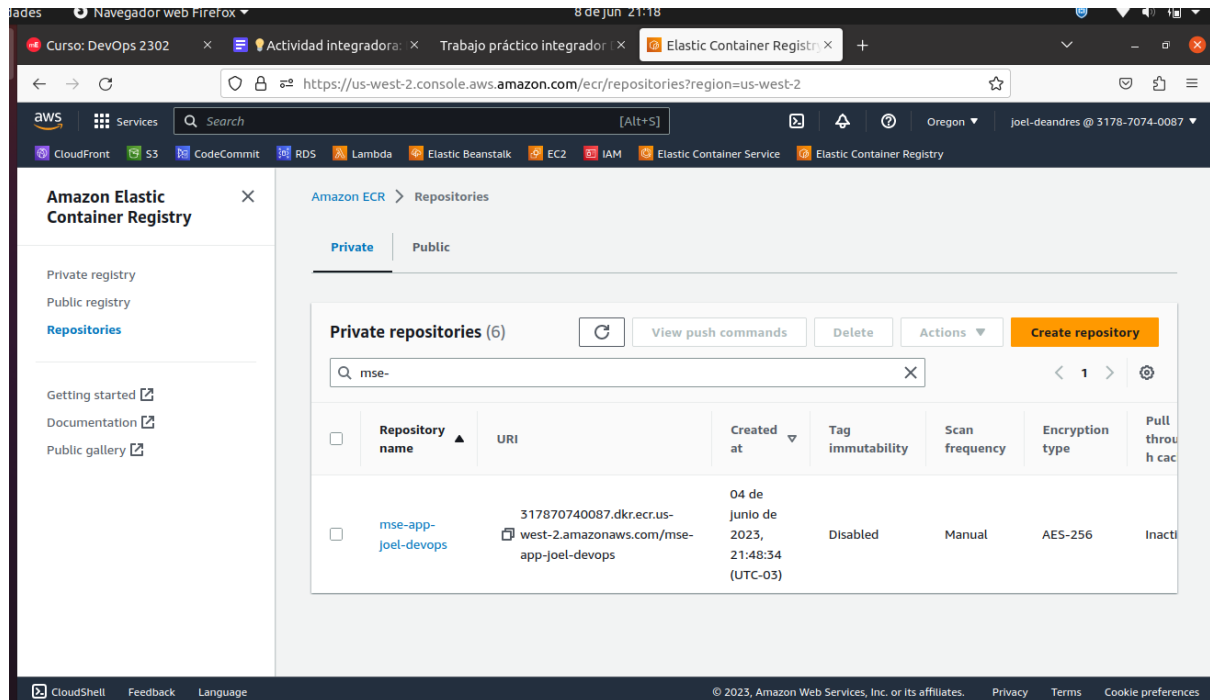
En el servicio ECR se creó el registry **mse-app-joel-devops**

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
```

Actividad integradora: Docker



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 imágenes 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
```

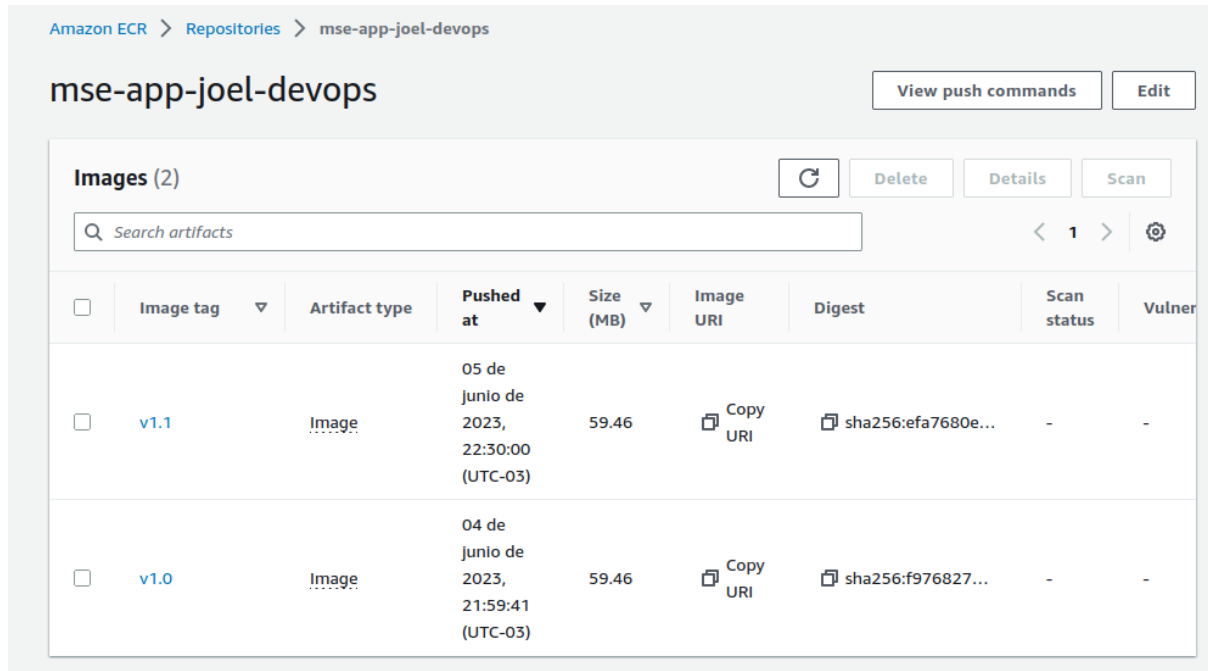
```
docker push \
${AWS_ACCOUNT_ID}.dkr.ecr.${AWS_REGION}.amazonaws.com/mse-app-joel-devops:v1.1
```

Actividad integradora: Docker

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.



Amazon ECR > Repositories > mse-app-joel-devops

mse-app-joel-devops [View push commands](#) [Edit](#)

Images (2) [Refresh](#) [Delete](#) [Details](#) [Scan](#)

<input type="checkbox"/>	Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest	Scan status	Vulner
<input type="checkbox"/>	v1.1	Image	05 de junio de 2023, 22:30:00 (UTC-03)	59.46	Copy URI	sha256:efa7680e...	-	-
<input type="checkbox"/>	v1.0	Image	04 de junio de 2023, 21:59:41 (UTC-03)	59.46	Copy URI	sha256:f976827...	-	-

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.

💡 Actividad integradora: Docker

Security Groups (1/2) Info

Filter security groups

search: mse-oper-joel X search: mse-web-joel X Clear filters

	Name	Security group ID	Security group name	VPC ID	Description
<input checked="" type="checkbox"/>	mse-oper-joel	sg-02cc97cfea36164eb	mse-oper-joel		Grupo de seguridad
<input type="checkbox"/>	mse-web-joel	sg-05bfb597aa7d6c3f2	mse-web-joel		grupo de seguridad

sg-02cc97cfea36164eb - mse-oper-joel

Details Inbound rules Outbound rules Tags

Inbound rules (3)

Filter security group rules

	Name	Security group rule...	IP version	Type	Protocol
<input type="checkbox"/>	-	src-0345db43b27d7e...	IPv4	SSH	TCP

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.

https://us-west-2.console.aws.amazon.com/ecs/v2/clusters/mse-ecs-cluster-joel/services?region=us-west-2

Amazon Elastic Container Service > Clusters > mse-ecs-cluster-joel > Services

mse-ecs-cluster-joel

Update cluster Delete cluster

Cluster overview

ARN	Status	CloudWatch monitoring	Registered container instances
mse-ecs-cluster-joel	Active	Default	2

Services

Draining

-

Active

1

Tasks

Pending

-

Running

1

Services Tasks Infrastructure Metrics Scheduled tasks Tags

En TASK se creó **mse-ecs-td-joel** y el VOLUMEN **mse-db_data**

Actividad integradora: Docker

The screenshot shows the AWS Management Console for the 'mse-ecs-td-joe1' task definition. The breadcrumb navigation is 'Amazon Elastic Container Service > Task definitions > mse-ecs-td-joe1 > Revision 1 > Containers'. The task definition name is 'mse-ecs-td-joe1:1'. There are buttons for 'Deploy', 'Actions', and 'Create new revision'. The 'Overview' tab is selected, showing the following details:

ARN	Status	Time created	App environment
arn:aws:ecs:us-west-2:317870740087:task-definition/mse-ecs-td-joe1:1	ACTIVE	7/6/2023, 04:00:25 UTC	EC2
Task role	Task execution role	Operating system/Architecture	Network mode
-	-	-	bridge

Se crearon Containers instances **mse-mongo-joe1** y **mse-app-joe1** para instalar mongo:4.2.23 en **mse-mongo-joe1** y la aplicación en **mse-app-joe1**

Containers Info						
Container name	Image	Private reg...	Essential	CPU	Memory	GPU
mse-mongo-joe1	mongo:...	-	Yes	0	.5 GB	-
mse-app-joe1	317870...	-	Yes	0	1 GB	-

En el servicio de EC2 se creo un Load Balancer con el nombre **mse-app-lb-joe1**

The screenshot shows the AWS Management Console for the 'mse-app-lb-joe1' load balancer in the EC2 console. The breadcrumb navigation is 'EC2 > Load balancers'. The 'Load balancers (1/1)' section shows a table with one entry:

Name	DNS name	State	VPC ID	Availability Zones
mse-app-lb-joe1	mse-app-lb-joe1-2021884...	Active	vpc-8edee5ea	2 Availability Zones

Below the table, there is a summary card for the 'Load balancer: mse-app-lb-joe1'.

y un Target Group con el nombre **mse-ecs-tg-joe1**

Actividad integradora: Docker

https://us-west-2.console.aws.amazon.com/ec2/home?region=us-west-2#TargetGroups:v=3;search=

Search [Alt+S]

CodeCommit RDS Lambda Elastic Beanstalk EC2 IAM Elastic Container Service Elastic Container Reg

EC2 > Target groups

Target groups (1/1) Info

Find resources by attribute or tag

joel X Clear filters

<input checked="" type="checkbox"/>	Name	ARN	Port	Protocol
<input checked="" type="checkbox"/>	mse-ecs-tg-joel	arn:aws:elasticloadbalanci...	8080	HTTP

Target group: mse-ecs-tg-joel

Details Targets Monitoring Health checks Attributes Tags

Details

arn:aws:elasticloadbalancing:us-west-2:317870740087:targetgroup/mse-ecs-tg-joel/74eb993ad8ca5571

Target type	Protocol : Port	Protocol version
Instance	HTTP: 8080	HTTP1

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



Actividad integradora: Docker

https://us-west-2.console.aws.amazon.com/ecs/v2/clusters/mse-ecs-cluster-joel/services?region=us-west-2

[Alt+S] Oregon joel-deandres

RDS Lambda Elastic Beanstalk EC2 IAM Elastic Container Service Elastic Container Registry

ARN: mse-ecs-cluster-joel Status: Active CloudWatch monitoring: Default Registered container instances: 2

Services

Draining: -

Active: 1

Tasks

Pending: -

Running: 1

Services (1/1) Info

Filter services by value All launch types All service types

<input checked="" type="checkbox"/>	Service name	Status	ARN	Service...	Deployment
<input checked="" type="checkbox"/>	mse-ecs-sv-joel	Active	arn:aws:ecs...	REPLICA	

Se verificó que la tarea esté en ejecución

https://us-west-2.console.aws.amazon.com/ecs/v2/clusters/mse-ecs-cluster-joel/tasks?region=us-west-2

[Alt+S] Oregon joel-deandres @ 31

RDS Lambda Elastic Beanstalk EC2 IAM Elastic Container Service Elastic Container Registry

ARN: mse-ecs-cluster-joel Status: Active CloudWatch monitoring: Default Registered container instances: 2

Services

Draining: -

Active: 1

Tasks

Pending: -

Running: 1

Tasks (1)

Filter tasks by property or value Running tasks All launch types

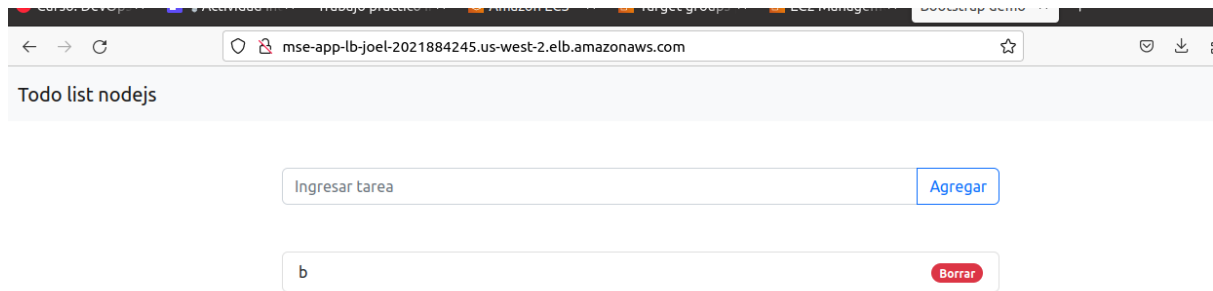
<input type="checkbox"/>	Task	Last status	Desired st...	T...	Re...	Health sta...	Started at
<input type="checkbox"/>	d9edff...	Running	Running	mse...	1	Unknown	2 days ago

Actividad integradora: Docker

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

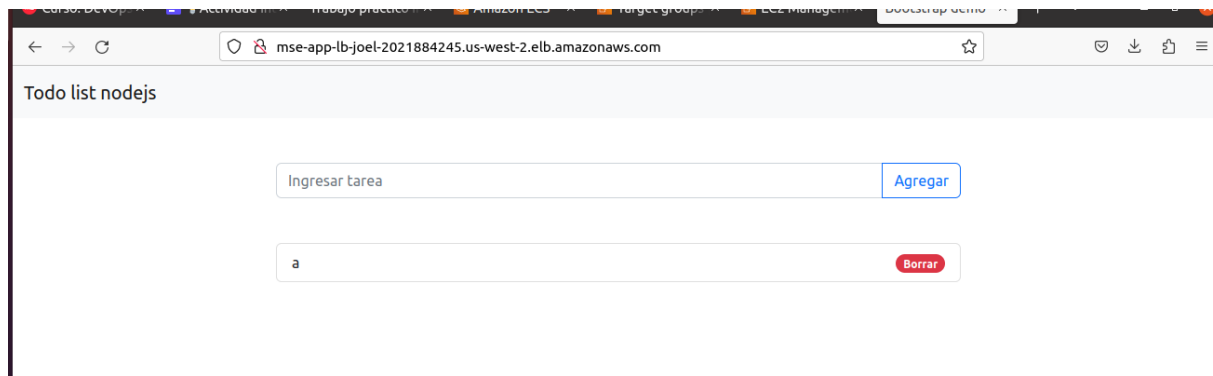
URL:

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



The screenshot shows a web browser window with the URL `mse-app-lb-joel-2021884245.us-west-2.elb.amazonaws.com`. The page title is "Todo list nodejs". Below the title, there is a text input field labeled "Ingresar tarea" and a blue button labeled "Agregar". Below the input field, there is a list item "b" with a red button labeled "Borrar" next to it.

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



The screenshot shows the same web browser window as before, but the list item "b" has been replaced by "a". The "Ingresar tarea" input field and the "Agregar" button are still present.

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.