

# PRACTICA 4

COMPUTACIÓN  
EN LA NUBE

# Contenido

INTRODUCCIÓN .....	2
ACTIVIDADES .....	3
1. Crear un contenedor con docker que tenga una aplicación que permita comprobar su funcionamiento (e.g. una pagina web) .....	3
2. Crear un repositorio en ECR y subir el contenedor creado en el paso 1 .....	5
3. Desplegar el contenedor usando ECS .....	7
4. Desplegar el contenedor usando Fargate y comparar la experiencia.....	14
ESTIMACIÓN DE GASTOS.....	18

# INTRODUCCIÓN

A lo largo de esta práctica se va a tener una primera toma de contacto con el sistema Docker, para la creación y administración de contenedores, se trabajará con el sistema de repositorios y a su vez se utilizarán los clústeres EC2 y Fargate de AWS.

# ACTIVIDADES

1. Crear un contenedor con docker que tenga una aplicación que permita comprobar su funcionamiento (e.g. una pagina web)

En primer lugar se crean ficheros en local para poder lanzar un servidor web donde aparecerá el contenido “Practica 4 CN”.

```
nikhil@DESKTOP-M2DV3PQ: ~  
const http = require('http');  
  
const hostname = '0.0.0.0';  
  
const server = http.createServer((req, res) => {  
  res.statusCode = 200;  
  res.setHeader('Content-Type', 'text/plain');  
  res.end('Practica 4 CN');  
});  
  
server.listen(5000, hostname, () => {  
  console.log('El servidor se esta ejecutando');  
});
```

Se debe configurar el dockerfile de la siguiente manera:

```
nikhil@DESKTOP-M2DV3PQ: ~  
FROM node:alpine  
COPY index.js /index.js  
EXPOSE 5000  
CMD ["node", "index.js"]
```

Para crear la imagen que se va a utilizar para el contenedor se debe ejecutar el siguiente comando:

`docker build -t practica4`

```
nikhil@DESKTOP-M2DV3PQ:~/practica4_cn$ docker build -t practica4 .  
[+] Building 30.1s (7/7) FINISHED  
=> [internal] load build definition from Dockerfile 0.0s  
=> => transferring dockerfile: 110B 0.0s  
=> [internal] load .dockerignore 0.0s  
=> => transferring context: 2B 0.0s  
=> [internal] load metadata for docker.io/library/node:alpine 3.0s  
=> [auth] library/node:pull token for registry-1.docker.io 0.0s  
=> [1/2] FROM docker.io/library/node:alpine@sha256:80844b6643f239c87fcea51e6540eeb054fc7114d979703770ec7525 26.8s  
=> => resolve docker.io/library/node:alpine@sha256:80844b6643f239c87fcea51e6540eeb054fc7114d979703770ec7525 0.0s  
=> => sha256:80844b6643f239c87fcea51e6540eeb054fc7114d979703770ec75250dc03b 1.43kB / 1.43kB 0.0s  
=> => sha256:dae8ae40ed1077dfa383fb0c04a3d3bb8e6360e03147dd3ee963d62ac2275346 1.16kB / 1.16kB 0.0s  
=> => sha256:9b78801b40588ad8ab8db899135501465b0a2d2519ccda04d8f36943582c9ac2 6.44kB / 6.44kB 0.0s  
=> => sha256:bfebca31f7556839677aca8626941ec4be0d5e2a1a59f1bd991807828de37167 47.57MB / 47.57MB 25.2s  
=> => sha256:cc0056ab0c4160f34cd7046016f9aa6d1d14c206f61768b34efa69c45c38a0cb 2.35MB / 2.35MB 1.7s  
=> => sha256:6e25476b6324255c964f6b86e587d867e79046e94933123d0f1312dbddfe87b7 453B / 453B 0.6s  
=> => extracting sha256:bfebca31f7556839677aca8626941ec4be0d5e2a1a59f1bd991807828de37167 1.3s  
=> => extracting sha256:cc0056ab0c4160f34cd7046016f9aa6d1d14c206f61768b34efa69c45c38a0cb 0.1s  
=> => extracting sha256:6e25476b6324255c964f6b86e587d867e79046e94933123d0f1312dbddfe87b7 0.0s  
=> [2/2] WORKDIR /app 0.2s  
=> => exporting to image 0.0s  
=> => exporting layers 0.0s  
=> => writing image sha256:f0f6bd4512efcf96ea490e60eca6b6c8f86edf986942317f36b1a6615382752e 0.0s  
=> => naming to docker.io/library/practica4 0.0s  
  
Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them  
nikhil@DESKTOP-M2DV3PQ:~/practica4_cn$
```

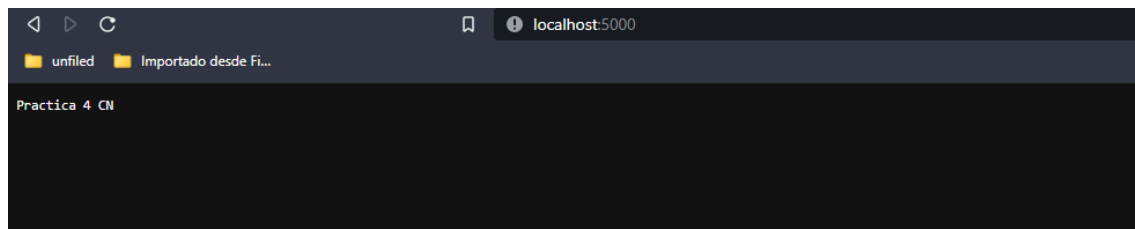
```

nikhil@DESKTOP-M2DV3PQ:~$ vim index.js
nikhil@DESKTOP-M2DV3PQ:~$ vim Dockerfile
nikhil@DESKTOP-M2DV3PQ:~$ docker build -t practica4 .
[+] Building 1.2s (7/7) FINISHED
=> [internal] load build definition from Dockerfile                                0.0s
=> => transferring dockerfile: 37B                                                0.0s
=> [internal] load .dockerignore                                                  0.0s
=> => transferring context: 2B                                                    0.0s
=> [internal] load metadata for docker.io/library/node:alpine                  1.0s
=> [internal] load build context                                                  0.0s
=> => transferring context: 346B                                                  0.0s
=> CACHED [1/2] FROM docker.io/library/node:alpine@sha256:80844b6643f239c87fcea51e6540eeb054fc7114d979703770 0.0s
=> [2/2] COPY index.js /index.js                                                0.0s
=> exporting to image                                                            0.0s
=> => exporting layers                                                            0.0s
=> => writing image sha256:dba89e2bf0e966905dc2f1fb75848bbbe9e2ada7bf57cb2836716e90bdbb5e 0.0s
=> => naming to docker.io/library/practica4                                     0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
nikhil@DESKTOP-M2DV3PQ:~$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
practica4     latest   dba89e2bf0e9   9 seconds ago  171MB
nikhil@DESKTOP-M2DV3PQ:~$ docker run -p 5000:5000 dba89e2bf0e9
El servidor se esta ejecutando

```

Una vez creada la imagen, se debe ejecutar **docker run -p 5000:5000 número\_id\_imagen** para poder crear el contenedor. Si se ha creado y ejecutado correctamente sale el mensaje programado “El servidor se está ejecutando”



## 2. Crear un repositorio en ECR y subir el contenedor creado en el paso 1

Se introducen las credenciales que se han copiado desde el laboratorio en un fichero con nombre *credentials* en la carpeta oculta *.aws*.

```
nikhil@DESKTOP-M2DV3PQ: ~  
nikhil@DESKTOP-M2DV3PQ:~/.aws$ ls  
nikhil@DESKTOP-M2DV3PQ:~/.aws$ touch credentials  
nikhil@DESKTOP-M2DV3PQ:~/.aws$ vim credentials  
nikhil@DESKTOP-M2DV3PQ:~/.aws$ cd ..  
nikhil@DESKTOP-M2DV3PQ:~$ aws configure  
  
Unable to parse config file: /home/nikhil/.aws/credentials  
nikhil@DESKTOP-M2DV3PQ:~$ cd .aws  
nikhil@DESKTOP-M2DV3PQ:~/.aws$ vim credentials  
nikhil@DESKTOP-M2DV3PQ:~/.aws$ cd ..  
nikhil@DESKTOP-M2DV3PQ:~$ aws configure  
AWS Access Key ID [*****B7YS]: ASIA2LP776FPAMC2B7YS  
AWS Secret Access Key [*****KSH4]: CkXAmplWwXNVjU1xMtFZAf8+mWVdh2lmFEG4KSH4  
Default region name [None]: us-east-1  
Default output format [None]: text  
nikhil@DESKTOP-M2DV3PQ:~$
```

A continuación se crea el repositorio en la plataforma AWS.

### Crear repositorio

#### Configuración general

**Configuración de visibilidad** [Info](#)  
Elija la configuración de visibilidad para el repositorio.

☒ **Privado**  
El acceso se administra mediante los permisos de las políticas de IAM y de repositorio.

☐ **Público**  
Visible y accesible públicamente para la extracción de imágenes.

**Nombre del repositorio**  
Proporcione un nombre conciso. Un desarrollador debe poder identificar el contenido del repositorio por el nombre.

711890825566.dkr.ecr.us-east-1.amazonaws.com/

9 de 256 caracteres máximos (2 mínimos). The name must start with a letter and can only contain lowercase letters, numbers, hyphens, underscores, periods and forward slashes.

**Inmutabilidad de etiqueta** [Info](#)  
Habilite la inmutabilidad de etiquetas para evitar que las etiquetas de imagen se sobrescriban en posteriores envíos de imágenes con la misma etiqueta. Deshabilite la inmutabilidad de etiquetas para permitir que se sobrescriban las etiquetas de imagen.

☐ **Desactivado**

[i](#) Una vez que se crea un repositorio, la configuración de visibilidad del repositorio no se puede cambiar.

## Comandos de envío para practica4



macOS / Linux

Windows

Asegúrese de tener instalada la versión más reciente de AWS CLI y Docker. Para obtener más información, consulte [Empezar a utilizar Amazon ECR](#).

Siga los siguientes pasos a fin de autenticar y enviar una imagen a su repositorio. Para obtener métodos de autenticación de registro adicionales, incluido el ayudante de credenciales de Amazon ECR, consulte [Autenticación del registro](#).

1. Recupere un token de autenticación y autentique su cliente de Docker en el registro.

Utilice AWS CLI:

```
aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 711890825566.dkr.ecr.us-east-1.amazonaws.com
```

Nota: Si recibe un error al utilizar AWS CLI, asegúrese de tener instaladas las últimas versiones de AWS CLI y Docker.

2. Cree una imagen de Docker con el siguiente comando. Para obtener información sobre cómo crear un archivo de Docker desde cero, consulte las instrucciones [aquí](#). Puede omitir este paso si ya se creó la imagen:

```
docker build -t practica4 .
```

3. Cuando se complete la creación, etiquete la imagen para poder enviarla a este repositorio:

```
docker tag practica4:latest 711890825566.dkr.ecr.us-east-1.amazonaws.com/practica4:latest
```

4. Ejecute el siguiente comando para enviar esta imagen al repositorio de AWS recién creado:

```
docker push 711890825566.dkr.ecr.us-east-1.amazonaws.com/practica4:latest
```

Cerrar

```
nikhil@DESKTOP-M2DV3PQ: ~
nikhil@DESKTOP-M2DV3PQ:~$ aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 711890825566.dkr.ecr.us-east-1.amazonaws.com
Login Succeeded

Logging in with your password grants your terminal complete access to your account.
For better security, log in with a limited-privilege personal access token. Learn more at https://docs.docker.com/go/access-tokens/
nikhil@DESKTOP-M2DV3PQ:~$ docker tag practica4:latest 711890825566.dkr.ecr.us-east-1.amazonaws.com/practica4:latest
nikhil@DESKTOP-M2DV3PQ:~$ docker push 711890825566.dkr.ecr.us-east-1.amazonaws.com/practica4:latest
The push refers to repository [711890825566.dkr.ecr.us-east-1.amazonaws.com/practica4]
1cc53860842d: Pushed
5957ac8a40f1: Pushed
7ae7e09dbb66: Pushed
a3ecf5a1df9e: Pushed
e5e13b0c77cb: Pushed
latest: digest: sha256:8d9de71907dd72cedff9382284f42eca1db4668eadd19d4cb2b7e307c7a37c9 size: 1365
nikhil@DESKTOP-M2DV3PQ:~$
```

Una vez que se ha creado el repositorio se ejecutan las respectivas instrucciones para poder realizar un push del contenido.

### 3. Desplegar el contenedor usando ECS

Para desplegar un contenedor ECS se van a crear 2 recursos, el primero es un clúster EC2 y lo segundo es su respectiva tarea.

#### Crear clúster

##### Paso 1: Seleccionar plantilla de clúster

Paso 2: Configurar el clúster

##### Seleccionar plantilla de clúster

Las siguientes plantillas de clúster están disponibles para simplificar la creación de clústeres. La configuración y las integraciones adicionales se pueden agregar más adelante.

**Solo redes**  
Recursos que se van a crear:  
Clúster  
VPC (opcional)  
Subredes (opcional)  
  
Para utilizar con AWS Fargate (Windows/Linux) o con capacidad de instancia externa.

**EC2 Linux + redes**  
Recursos que se van a crear:  
Clúster  
VPC  
Subredes  
Grupo de escalado automático con la AMI de Linux

**EC2 Windows + redes**  
Recursos que se van a crear:  
Clúster  
VPC  
Subredes  
Grupo de escalado automático con la AMI de Windows

EC2 > Grupos de seguridad > Crear grupo de seguridad

#### Crear grupo de seguridad

Información

Un grupo de seguridad actúa como un firewall virtual para que la instancia controle el tráfico de entrada y salida. Para crear un nuevo grupo de seguridad, complete los campos siguientes.

##### Detalles básicos

Nombre del grupo de seguridad [Información](#)

pg\_grupo1

El nombre no se puede editar después de su creación.

Descripción [Información](#)

Permite el acceso SSH a los desarrolladores

VPC [Información](#)

vpc-05d6cc932cb5cf9e

##### Reglas de entrada [Información](#)

Tipo <a href="#">Información</a>	Protocolo <a href="#">Información</a>	Intervalo de puertos <a href="#">Información</a>	Origen <a href="#">Información</a>	Descripción: opcional <a href="#">Información</a>	
TCP personalizado	TCP	5000	Anywhere-IPv4	Q	Eliminar
				0.0.0.0/0	
TCP personalizado	TCP	5000	Anywhere-IPv6	Q	Eliminar
				:::0	
Todo el tráfico	Todo	Todo	Personalizada	Q	Eliminar
				sg-0199b1342f684464c	
Agregar regla					

Se crea el grupo de seguridad que se utilizará posteriormente.



Paso 1: Seleccionar plantilla de clúster

Paso 2: Configurar el clúster

## Configurar el clúster

Nombre del clúster\*

cluster-ec2



☐ Crear un clúster vacío

## Configuración de la instancia

Modelo de aprovisionamiento

☒ Instancia bajo demanda

Con las instancias bajo demanda, se paga por la capacidad de cómputo por hora, sin compromisos a largo plazo ni pagos por adelantado.

☐ Spot

Las instancias de spot de Amazon EC2 permiten aprovechar la capacidad de EC2 que no se utiliza en la nube de AWS. Las instancias de spot están disponibles con un descuento de hasta el 90 % en comparación con los precios bajo demanda. [Más información](#)

Tipo de instancia de EC2\*

t3.micro



☐ Ingrese manualmente el tipo de instancia deseada

Cantidad de instancias\*

1



ID de la AMI de EC2\*

Amazon Linux 2 AMI [ami-0fe5f...



Tamaño del volumen de EBS raíz (GiB)

30



Par de claves

Template



No podrá acceder mediante SSH a las

## Redes

Configure la VPC para que las instancias de contenedor la utilicen. Una VPC es una porción aislada de la nube de AWS rellena por objetos de AWS, como las instancias de Amazon EC2. Puede elegir una VPC existente o crear una nueva con este asistente.

VPC

vpc-05d6ccc932cb5cf9e...



Compruebe la estructura de [vpc-05d6ccc932cb5cf9e](#) en la consola de Amazon EC2.

Subredes

subnet-048b5d8cfa6aa953a  
(172.31.16.0/20) - us-east-1a  
asignar ipv6 en la creación: Desactivado



Seleccionar una subred...

Asignar automáticamente una IP pública

Habilitado



Grupo de seguridad

sg-032f3b81a98ca3291 ...



Reglas correspondientes a [sg-032f3b81a98ca3291](#) en la consola de EC2.

Se utiliza el grupo de seguridad que se creó anteriormente.

9

## Instancia de contenedor Rol de IAM

El agente de contenedores de Amazon ECS realiza llamadas a las acciones de la API de Amazon ECS en su nombre, por lo que las instancias de contenedor que ejecutan el agente requieren la política y el rol de IAM `ecsInstanceRole` para que el servicio sepa que el agente le pertenece a usted. Si aún no tiene el `ecsInstanceRole`, podemos crear uno para usted.

Rol de IAM de instancia de contenedor

LabRole



Para que las instancias de contenedor reciban el nuevo formato de ARN e ID de recurso, el usuario raíz debe aceptar el rol de IAM de la instancia de contenedor. Acepte e inténtelo de nuevo.

## Etiquetas

Clave

Valor

Agregar clave

Agregar valor

## Información de contenedores de CloudWatch

Información sobre contenedores de CloudWatch es una solución de supervisión y resolución de problemas para aplicaciones y microservicios en contenedores. Recopila, agrega y resume la utilización de los recursos de computación, como la CPU, la memoria, el disco y la red, así como información de diagnóstico, como los errores de reinicio de los contenedores, a fin de ayudar a aislar los problemas de los clústeres y resolverlos rápidamente. [Más información](#)

Información de contenedores de CloudWatch

☐ Habilitar Información sobre contenedores

\*Obligatorio

Cancelar

Anterior

Crear

## Estado del lanzamiento

Las instancias de contenedor están en proceso de lanzamiento y pueden tardar unos minutos hasta que estén en estado de ejecución y listas para acceder. Las horas de uso de las nuevas instancias de contenedor se inician inmediatamente y continúan acumulándose hasta que se detengan o terminen.

[Volver](#) [Ver clúster](#)

Estado de ECS - 3 de 3 completos **cluster-ec2**

✓ Clúster de ECS  
El clúster de ECS cluster-ec2 se creó correctamente.

✓ Política de IAM de la instancia de ECS  
Se asoció correctamente la política de IAM correspondiente al rol `LabInstanceProfile`.

✓ Pila de CloudFormation  
La pila de CloudFormation `EC2ContainerService-cluster-ec2` y sus recursos se crearon correctamente.

Recursos del clúster

Tipo de instancia	t3.micro
Cantidad deseada de instancias	1
Par de claves	Templata
ID de IAM de ECS	arn:aws:iam::080958ca
VPC	vpc-0566cc32d45c5f9e
Subredes	subnet-048b5d8cfafaa953a
Zonas de disponibilidad de la VPC	us-east-1a, us-east-1b, us-east-1c, us-east-1d, us-east-1e, us-east-1f
Grupo de seguridad	sg-0325ba1a9f6a43291
Configuración de lanzamiento	EC2ContainerService-cluster-ec2-EcsInstanceProfile-771q
Grupo de escalado automático	EC2ContainerService-cluster-ec2-EcsInstanceAsp-1W7F3WSZBNQ3

A partir de aquí se lleva a cabo la definición de la tarea, como el clúster es de EC2, la tarea tiene que seguir el mismo formato.

### Crear una nueva definición de tarea

Paso 1: Seleccione la compatibilidad con el tipo de lanzamiento  
Paso 2: Configurar definiciones de tareas y contenedores

#### Seleccione la compatibilidad con el tipo de lanzamiento

Seleccione el tipo de lanzamiento con el que desea que la definición de la tarea sea compatible en función del lugar en el que desea lanzar la tarea.

**FARGATE**  
  
Precio basado en el tamaño de la tarea  
Requiere el modo de red awsvpc  
Infraestructura administrada por AWS, sin instancias de Amazon EC2 que administrar

**EC2**  
  
Precio basado en el uso de recursos  
Varios modos de red disponibles  
Infraestructura autoadministrada mediante instancias de Amazon EC2

**EXTERNO**  
  
Precio basado en horas de instancia y cargos adicionales por otros servicios de AWS utilizados  
Infraestructura local autoadministrada con ECS Anywhere

\*Obligatorio

Cancelar

Paso siguiente

### Configurar definiciones de tareas y contenedores

Una definición de tarea especifica qué contenedores se incluyen en la tarea y el modo en que interactúan entre sí. También es posible especificar los volúmenes de datos que los contenedores van a utilizar. [Más información](#)

Nombre de la definición de tarea\* tarea1\_p4 ⓘ

Requiere compatibilidades\* EC2

Rol de la tarea Seleccionar un rol... ⓘ

Rol de IAM opcional que las tareas pueden utilizar para realizar solicitudes a la API a los servicios de AWS autorizados. Cree un rol de tarea de Amazon Elastic Container Service en la [consola de IAM](#)

Modo de red <default> ⓘ

Si elige , ECS iniciará el contenedor con el modo de red predeterminado de Docker, que es Bridge en Linux y NAT en Windows. Las tareas de Windows admiten los modos de red y awsvpc.

## Tamaño de la tarea



El tamaño de la tarea permite especificar un tamaño fijo para la tarea. El tamaño de la tarea es necesario para las tareas que utilizan el tipo de lanzamiento Fargate y es opcional para los tipos de lanzamiento EC2 o Externo. La configuración de la memoria a nivel de contenedor es opcional cuando se establece el tamaño de la tarea. El tamaño de la tarea no es compatible con los contenedores de Windows.

Memoria de la tarea (MiB)

500

La cantidad de memoria (en MiB) utilizada por la tarea. Se puede expresar como un número entero mediante MiB, por ejemplo 1024, o como una cadena mediante GB, por ejemplo "1GB" o "1 gb".

CPU de la tarea (unidad)

1 vcpu

La cantidad de unidades de CPU utilizadas por la tarea. Se puede expresar como un número entero mediante unidades de CPU, por ejemplo 1024, o como una cadena mediante vCPU, por ejemplo "1 vCPU" o "1 vcpu".

### Asignación máxima de memoria de tarea para la reserva de memoria del contenedor



### Asignación máxima de la CPU de la tarea para los contenedores



## Definiciones de contenedores



Agregar contenedor

Nombre del...	Imagen	Límites de ...	Unidades ...	GPU	Acelerador ...	Esencial ...	
contene...	7118908255...	--/--				true	✕

### Agregar contenedor



#### ▼ Estándar

Nombre del contenedor\*

contenedor\_ec2\_p4



Imagen\*

711890825566.dkr.ecr.us-east-1.amazonaws.com/practica4:latest



Autenticación de repositorios privados\*



Límites de memoria (MiB)\*

Límite est... ▼

128



[+ Agregar límite flexible](#)

Define límites de memoria estrictos o flexibles en MiB para el contenedor. Los límites estrictos y flexibles corresponden a los parámetros "memory" y "memoryReservation", respectivamente, en las definiciones de las tareas. ECS recomienda 300-500 MiB como punto de partida para las aplicaciones web.

Asignaciones de puertos

Puerto del host

5000

Puerto del contenedor

5000

Protocolo

tcp ▼



[+ Agregar asignación de puertos](#)

Cabe destacar la asignación de puertos, por ello se les asigna el puerto 5000 como se hizo anteriormente.

Una vez terminada la definición de la tarea, se procede a realizar su ejecución, que no es más que cargar la tarea creada y ejecutarla.

### Ejecutar tarea

Seleccione el clúster en el que se va a ejecutar la definición de la tarea y la cantidad de copias de esa tarea que se van a ejecutar. Para aplicar anulaciones de contenedor o dirigirse a instancias de contenedor concretas, haga clic en Opciones avanzadas.

Tipo de lanzamiento ☐ FARGATE ☒ EC2 ☐ EXTERNAL ⓘ

[Cambiar a la estrategia de proveedor de capacidad](#) ⓘ

Definición de tarea

Familia

tarea1\_p4

[Ingrese un valor](#)

Revisión

1 (latest)

Clúster

cluster-ec2

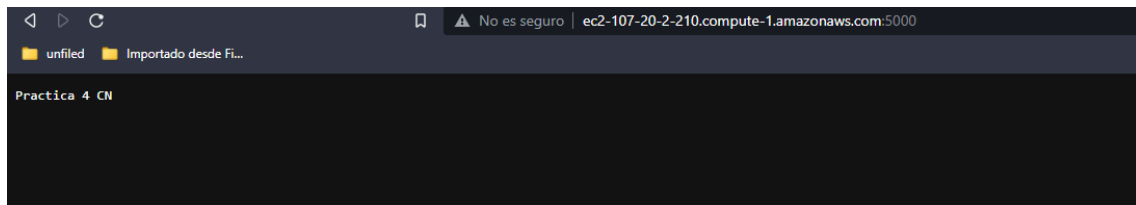
Cantidad de tareas

1

Grupo de tareas

ⓘ

Se puede observar el funcionamiento del contenedor ECS.



## 4. Desplegar el contenedor usando Fargate y comparar la experiencia

Para crear el contenedor usando Fargate es prácticamente lo mismo que el caso anterior solo que a la hora de crear el clúster se selecciona el de Solo Redes.

### Crear clúster

Paso 1: Seleccionar plantilla de clúster

Paso 2: Configurar el clúster

#### Seleccionar plantilla de clúster

Las siguientes plantillas de clúster están disponibles para simplificar la creación de clústeres. La configuración y las integraciones adicionales se pueden agregar más adelante.

**Solo redes**  
Recursos que se van a crear:  
Clúster  
VPC (opcional)  
Subredes (opcional)  
  
**Para utilizar con AWS Fargate (Windows/Linux) o con capacidad de instancia externa.**

**EC2 Linux + redes**  
Recursos que se van a crear:  
Clúster  
VPC  
Subredes  
Grupo de escalado automático con la AMI de Linux

**EC2 Windows + redes**  
Recursos que se van a crear:  
Clúster  
VPC  
Subredes  
Grupo de escalado automático con la AMI de Windows

\*Obligatorio

Cancelar

Paso siguiente

### Crear clúster

Paso 1: Seleccionar plantilla de clúster

Paso 2: Configurar el clúster

#### Configurar el clúster

Nombre del clúster\* cluster-fargate

#### Redes

Cree una nueva VPC para que el clúster la utilice. Una VPC es una porción aislada de la nube de AWS en la que se encuentran objetos de AWS, como las tareas de Fargate.

Crear VPC ☐ Crear una nueva VPC para este clúster

#### Etiquetas

Clave	Valor
<input type="text" value="Agregar clave"/>	<input type="text" value="Agregar valor"/>

#### Información de contenedores de CloudWatch

Información sobre contenedores de CloudWatch es una solución de supervisión y resolución de problemas para aplicaciones y microservicios en contenedores. Recopila, agrega y resume la utilización de los recursos de computación, como la CPU, la memoria, el disco y la red, así como información de diagnóstico, como los errores de reinicio de los contenedores, a fin de ayudar a aislar los problemas de los clústeres y resolverlos rápidamente. [Más información](#)

Información de contenedores de CloudWatch ☐ Habilitar Información sobre contenedores

\*Obligatorio

Cancelar

Anterior

Crear

A continuación se define la nueva tarea y en esta ocasión se escoge FARGATE.

### Crear una nueva definición de tarea

**Paso 1:** Seleccione la compatibilidad con el tipo de lanzamiento

Paso 2: Configurar definiciones de tareas y contenedores

#### Seleccione la compatibilidad con el tipo de lanzamiento

Seleccione el tipo de lanzamiento con el que desea que la definición de la tarea sea compatible en función del lugar en el que desea lanzar la tarea.

**FARGATE**  
  
Precio basado en el tamaño de la tarea  
Requiere el modo de red awsvpc  
Infraestructura administrada por AWS, sin instancias de Amazon EC2 que administrar

**EC2**  
  
Precio basado en el uso de recursos  
Varios modos de red disponibles  
Infraestructura autoadministrada mediante instancias de Amazon EC2

**EXTERNO**  
  
Precio basado en horas de instancia y cargos adicionales por otros servicios de AWS utilizados  
Infraestructura local autoadministrada con ECS Anywhere

\*Obligatorio

Cancelar

Paso siguiente

Agregar contenedor

▼ Estándar

Nombre del contenedor\*

contenedor2\_p4

?

Imagen\*

711890825566.dkr.ecr.us-east-1.amazonaws.com/practica4:latest

?

Autenticación de repositorios privados\*

☐

?

Límites de memoria (MiB)

Límite fle...

128

?

➕ Agregar límite estricto

Defina límites de memoria estrictos o flexibles en MiB para el contenedor. Los límites estrictos y flexibles corresponden a los parámetros "memory" y "memoryReservation", respectivamente, en las definiciones de las tareas.  
ECS recomienda 300-500 MiB como punto de partida para las aplicaciones web.

Asignaciones de puertos

Puerto del contenedor

5000

Protocolo

tcp

?

➕ Agregar asignación de puertos

Las asignaciones de puertos de host no son válidas cuando el modo de red de una definición de tarea es host o awsvpc. Para especificar diferentes asignaciones de puertos de host y de contenedor, elija el modo de red Bridge.

▼ Configuración avanzada de contenedores

COMPROBACIÓN DE ESTADO

Comando

CMD-SHELL, curl -f http://localhost/ || exit 1

?

\* Obligatorio

Cancelar

Agregar

15



## Crear una nueva definición de tarea

Paso 1: Seleccione la compatibilidad con el tipo de lanzamiento

Paso 2: Configurar definiciones de tareas y contenedores

### Configurar definiciones de tareas y contenedores

Una definición de tarea especifica qué contenedores se incluyen en la tarea y el modo en que interactúan entre sí. También es posible especificar los volúmenes de datos que los contenedores van a utilizar. [Más información](#)

Nombre de la definición de tarea\*  ⓘ

Requiere compatibilidades\* FARGATE

Rol de la tarea  ⓘ  
Rol de IAM opcional que las tareas pueden utilizar para realizar solicitudes a la API a los servicios de AWS autorizados. Cree un rol de tarea de Amazon Elastic Container Service en la [consola de IAM](#) ⓘ

Modo de red  ⓘ  
Si elige , ECS iniciará el contenedor con el modo de red predeterminado de Docker, que es Bridge en Linux y NAT en Windows. Las tareas de Windows admiten los modos de red y awsvpc.

Familia de sistemas operativos  ⓘ

### Ejecución de tareas Rol de IAM

Este rol es requerido por las tareas para extraer imágenes de contenedores y publicar registros de contenedores en Amazon CloudWatch en su nombre. Si aún no tiene el rol `ecsTaskExecutionRole`, podemos crear uno para usted.

Rol de ejecución de tareas  ⓘ

### Tamaño de la tarea

El tamaño de la tarea permite especificar un tamaño fijo para la tarea. El tamaño de la tarea es necesario para las tareas que utilizan el tipo de lanzamiento Fargate y es opcional para los tipos de lanzamiento EC2 o Externo. La configuración de la memoria a nivel de contenedor es opcional cuando se establece el tamaño de la tarea. El tamaño de la tarea no es compatible con los contenedores de Windows.

Memoria de la tarea (GB)

La cantidad de memoria (en MiB) utilizada por la tarea. Se puede expresar como un número entero mediante MiB, por ejemplo 1024, o como una cadena mediante GB, por ejemplo "1GB" o "1 gb".

CPU de la tarea (vCPU)

La cantidad de unidades de CPU utilizadas por la tarea. Se puede expresar como un número entero mediante unidades de CPU, por ejemplo 1024, o como una cadena mediante vCPU, por ejemplo "1 vCPU" o "1 vcpu".

### Asignación máxima de memoria de tarea para la reserva de memoria del contenedor



### Asignación máxima de la CPU de la tarea para los contenedores



### Definiciones de contenedores

[Agregar contenedor](#)

Nombre del...	Imagen	Límites de ...	Unidades ...	GPU	Acelerador ...	Esencial ...	
contene...	7118908255...	--/--				true	✕

## Ejecutar tarea

Seleccione el clúster en el que se va a ejecutar la definición de la tarea y la cantidad de copias de esa tarea que se van a ejecutar. Para aplicar anulaciones de contenedor o dirigirse a instancias de contenedor concretas, haga clic en Opciones avanzadas.

Tipo de lanzamiento ☒ FARGATE ☐ EC2 ☐ EXTERNAL ⓘ

AWS Fargate se encuentra en proceso de migrar las cuotas de servicio de las actuales cuotas basadas en el recuento de tareas de Amazon ECS a cuotas basadas en vCPU. Para obtener más información, consulte las preguntas frecuentes sobre AWS Fargate.

[Cambiar a la estrategia de proveedor de capacidad](#) ⓘ

Familia de sistemas operativos Linux ▾

Definición de tarea Familia   
 ec2\_p4\_tarea2 ▾ [Ingrese un valor](#)   
 Revisión   
 1 (latest) ▾

Versión de la plataforma LATEST ▾ ⓘ

Clúster cluster-fargate ▾

Cantidad de tareas 1

Grupo de tareas   
  ⓘ

## VPC y grupos de seguridad

La VPC y los grupos de seguridad se pueden configurar cuando la definición de la tarea utiliza el modo de red awsvpc.

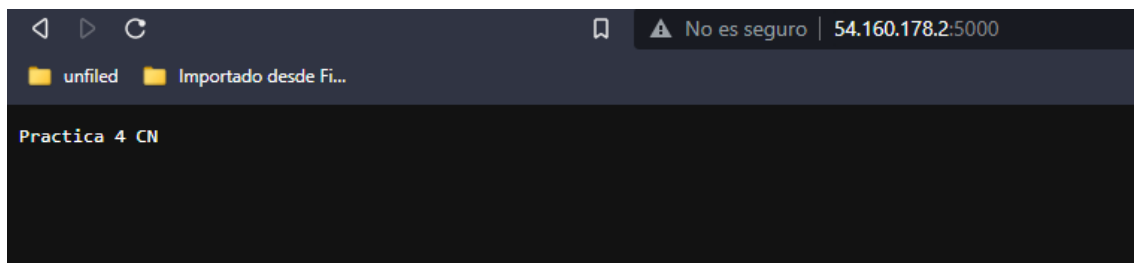
VPC del clúster\* vpc-05d6ccc932cb5cf9e (172.31.0.0/16) ⓘ

Subredes\* subnet-048b5d8cfa6aa953a (172.31.16.0/20) - us-east-1a   
 asignar ipv6 en la creación: Desactivado   
 ⓘ

Grupos de seguridad\* sg-032f3b81a98ca3291 [Editar](#) ⓘ

Asignar automáticamente una IP pública ENABLED ⓘ

Se puede observar que el clúster FARGATE funciona correctamente.



## ESTIMACIÓN DE GASTOS

Cuando se despliega el contenedor ECS se utiliza una instancia de tipo t3.micro. Esta a comparación con las otras prácticas es un poco más cara. El coste por cada hora constante de uso es de 0.0104\$/h. Esto juntado al amazon EBS supone un coste mensual de 11.27\$. Lo cual se traduce en 135.24\$ anuales.

Mientras que el despliegue del contenedor usando FARGATE tiene los siguientes gastos:

Región:	EE.UU. Este (Norte de Virginia) *
Precio	
por CPU virtual por hora	0,0138844 USD
por GB por hora	0,00152461 USD

Esto según la configuración que se ha utilizado se traduce en 3.05\$ mensuales o que es lo mismo 36,6\$ anuales.