



Práctica 4: Orquestación de Contenedores.

Albañiles Digitales: Cloud (Amazon Web Services)

Introducción

Como ya hemos visto en la teoría, hay muchas razones por las que orquestar los contenedores y crear un servicio. En esta práctica vais a desplegar un servicio desde la consola de AWS. La entrega final serán unas capturas del proceso, así que id guardando capturas de pantalla de como vais haciendo el proceso. El guión de la práctica indicará cuándo hacer las fotos.

Recordemos los pasos que hay que hacer:

1. Crear un Cluster para el servicio
2. Crear un tipo de tarea para el servicio
3. Crear un balanceador de carga (para saber a qué máquina apuntar)
4. Crear el servicio

Ahora vamos a ir paso por paso viendo cómo podemos hacerlo. En este caso **NO queremos ocuparnos de las instancias EC2**. Veremos cómo funcionan las **instancias Fargate**, para quitarnos un dolor de cabeza de encima.

1. Crear un Cluster

Recordemos que un **cluster es un conjunto de instancias/máquinas que trabajan conjuntamente con un objetivo**. Vamos a empezar creando ese grupo.

1. Entramos en “Elastic Container Service”

The screenshot shows the AWS Elastic Container Service (ECS) console. On the left, there's a sidebar with various AWS services like CloudWatch, Lambda, and S3. The main search bar at the top has 'elastic container service' typed into it. Below the search bar, there are two main sections: 'Servicios' and 'Características'. Under 'Servicios', there are three items: 'Elastic Container Service', 'Elastic Container Registry', and 'Elastic Transcoder'. Under 'Características', there are three items: 'Introducción', 'Definiciones de tareas', and 'Espacios de nombres'. A modal window titled 'Presentación de la búsqueda de recursos' is open, providing instructions for using the search feature. At the bottom of the page, there's a 'Publicaciones de blog' section with 266 results.

2. Veremos un botón que dice “Create Cluster”. Clicamos en él

The screenshot shows the 'Clusters' page under the 'Amazon Elastic Container Service' section. The left sidebar includes options like 'Cíusters', 'Amazon ECR', 'AWS Batch', and 'Documentación'. The main content area shows a table with one row labeled 'Sin clústeres' and 'No hay clústeres que mostrar'. At the top right of the main area, there is a prominent orange 'Crear clúster' button.

3. Como vamos a usar instancias AWS Fargate, clicamos en “Networking only”

Este botón o no existe o no lo encuentro o por default te lo dá así predeterminado.

4. Lo configuraremos de tal manera:

- a. Le ponemos un nombre. Por ejemplo “alba-digi-cluster”

Crear clúster Información

Un clúster de Amazon ECS agrupa tareas y servicios, y permite las configuraciones comunes y la capacidad compartida. Todas las tareas, los servicios y la capacidad deben pertenecer a un clúster.

Configuración del clúster

Nombre del clúster

El nombre del clúster debe tener entre 1 y 255 caracteres. Los caracteres válidos son a-z, A-Z, 0-9, guiones (-) y guiones bajos (_).

Espacio de nombres predeterminado - opcional

Seleccione el espacio de nombres para especificar un grupo de servicios que componen la aplicación. Puede sobrescribir este valor en el nivel de servicio.

Alba-digi-cluster

Infraestructura Información

Este clúster se configura automáticamente para AWS Fargate (sin servidor) con dos proveedores de capacidad. Agregue Instancias de Amazon EC2.

AWS Fargate (sin servidor)

Pago por uso. Utilícelo si tiene cargas de trabajo pequeñas, por lotes o por ráfagas o sin gastos de mantenimiento. El clúster tiene proveedores de capacidad Fargate y Fargate Spot de forma predeterminada.

Instancias de Amazon EC2

Configuraciones manuales. Utilícelas para grandes cargas de trabajo con demandas de recursos coherentes.

Amazon Lambda

Las instancias externas que utilizan **ECS Anywhere** se pueden registrar una vez finalizada la creación del clúster.

Monitoreo - opcional Información

CloudWatch Container Insights es una solución de supervisión y resolución de problemas para aplicaciones y microservicios en contenedores.

Cifrado - opcional

Elija las claves de KMS que utilizan las tareas que se ejecutan en este clúster para cifrar el almacenamiento.

Etiquetas - opcionales Información

Las etiquetas le ayudan a identificar y organizar los clústeres.

b. En infrastructure, selecciona 'AWS Fargate (serverless)'

Crear clúster Información

Este clúster se configura automáticamente para AWS Fargate (sin servidor) con dos proveedores de capacidad. Agregue Instancias de Amazon EC2.

AWS Fargate (sin servidor)

Pago por uso. Utilícelo si tiene cargas de trabajo pequeñas, por lotes o por ráfagas o sin gastos de mantenimiento. El clúster tiene proveedores de capacidad Fargate y Fargate Spot de forma predeterminada.

Instancias de Amazon EC2

Configuraciones manuales. Utilícelas para grandes cargas de trabajo con demandas de recursos coherentes.

Amazon Lambda

Las instancias externas que utilizan **ECS Anywhere** se pueden registrar una vez finalizada la creación del clúster.

Monitoreo - opcional Información

CloudWatch Container Insights es una solución de supervisión y resolución de problemas para aplicaciones y microservicios en contenedores.

Cifrado - opcional

Elija las claves de KMS que utilizan las tareas que se ejecutan en este clúster para cifrar el almacenamiento.

Etiquetas - opcionales Información

Las etiquetas le ayudan a identificar y organizar los clústeres.

La configuración de cuenta para la autorización del etiquetado de recursos está activada actualmente.

La acción `ecs:TagResource` es necesaria para etiquetar recursos de ECS.

No hay etiquetas asociadas al recurso.

Agregar etiqueta

Puede agregar 50 etiquetas más.

Cancelar **Crear**

c. En tags ponemos. Key: "Name" Value: "alba-digi-cluster"

▼ Infraestructura Información

Su clúster se configura automáticamente para AWS Fargate (sin servidor) con dos proveedores de capacidad. Agregue Instancias de Amazon EC2.

AWS Fargate (sin servidor)
Pago por uso. Utilícelo si tiene cargas de trabajo pequeñas, por lotes o por ráfagas o sin gastos de mantenimiento. El clúster tiene proveedores de capacidad Fargate y Fargate Spot de forma predeterminada.

Instancias de Amazon EC2
Configuraciones manuales. Utilícelas para grandes cargas de trabajo con demandas de recursos coherentes.

ⓘ Las instancias externas que utilizan **ECS Anywhere** se pueden registrar una vez finalizada la creación del clúster.

Sin servidor

► Monitoreo - opcional Información

CloudWatch Container Insights es una solución de supervisión y resolución de problemas para aplicaciones y microservicios en contenedores.

► Cifrado - opcional

Elja las claves de KMS que utilizan las tareas que se ejecutan en este clúster para cifrar el almacenamiento.

▼ Etiquetas - opcionales Información

Las etiquetas le ayudan a identificar y organizar los clústeres.

ⓘ La configuración de cuenta para la autorización del etiquetado de recursos está activada actualmente.
La acción ecs:TagResource es necesaria para etiquetar recursos de ECS.

Clave	Valor - opcional	Eliminar
Name	alba-digi-cluster	Eliminar

Agregar etiqueta

Puede agregar 49 etiquetas más.

d. NO clicamos en Enable Container Insights.

▼ Infraestructura Información

Su clúster se configura automáticamente para AWS Fargate (sin servidor) con dos proveedores de capacidad. Agregue Instancias de Amazon EC2.

AWS Fargate (sin servidor)
Pago por uso. Utilícelo si tiene cargas de trabajo pequeñas, por lotes o por ráfagas o sin gastos de mantenimiento. El clúster tiene proveedores de capacidad Fargate y Fargate Spot de forma predeterminada.

Instancias de Amazon EC2
Configuraciones manuales. Utilícelas para grandes cargas de trabajo con demandas de recursos coherentes.

ⓘ Las instancias externas que utilizan **ECS Anywhere** se pueden registrar una vez finalizada la creación del clúster.

▼ Monitoreo - opcional Información

CloudWatch Container Insights es una solución de supervisión y resolución de problemas para aplicaciones y microservicios en contenedores.

Seleccione el nivel de observabilidad que desea alcanzar con la información de contenedores
La información de contenedores está desactivada de forma predeterminada en la configuración de la cuenta. Puede anularlo aquí, a nivel de clúster.

Información de contenedores con observabilidad mejorada Opción recomendada
Proporciona métricas detalladas sobre el estado y el rendimiento a nivel de tarea y de contenedor, además de métricas agregadas a nivel de clúster y de servicio. Permite realizar análisis en profundidad más fáciles para aislar y solucionar problemas con mayor rapidez.

Container Insights
Proporciona métricas agregadas a nivel de clúster y de servicio. Puede realizar un análisis profundo con los análisis de la información de registros.

Desactivado
Proporciona únicamente métricas predeterminadas de CloudWatch.

► Cifrado - opcional

Elja las claves de KMS que utilizan las tareas que se ejecutan en este clúster para cifrar el almacenamiento.

▼ Etiquetas - opcionales Información

Las etiquetas le ayudan a identificar y organizar los clústeres.

ⓘ La configuración de cuenta para la autorización del etiquetado de recursos está activada actualmente.
La acción ecs:TagResource es necesaria para etiquetar recursos de ECS.

e. Le damos a create

Monitoreo - opcional Información

CloudWatch Container Insights es una solución de supervisión y resolución de problemas para aplicaciones y microservicios en contenedores.

Seleccione el nivel de observabilidad que desea alcanzar con la información de contenedores

La información de contenedores está desactivada de forma predeterminada en la configuración de la cuenta. Puede activarla aquí, a nivel de clúster.

Información de contenedores con observabilidad mejorada Opción recomendada

Proporciona métricas detalladas sobre el estado y el rendimiento a través de tareas y de contenedor, además de métricas agregadas a nivel de clúster y de servicio. Permite realizar análisis en profundidad más fáciles para aislar y solucionar problemas con mayor rapidez.

Container Insights

Proporciona métricas agregadas a nivel de clúster y de servicio. Puede realizar un análisis profundo con los análisis de la Información de registros.

Desactivado

Proporciona únicamente métricas predeterminadas de CloudWatch.

Cifrado - opcional

Elija las claves de KMS que utilizan las tareas que se ejecutan en este clúster para cifrar el almacenamiento.

Etiquetas - opcionales Información

Las etiquetas le ayudan a identificar y organizar los clústeres.

La configuración de cuenta para la autorización del etiquetado de recursos está activada actualmente.

La acción ecs:TagResource es necesaria para etiquetar recursos de ECS.

Clave	Valor - opcional
Name	alba-digi-cluster

Agregar etiqueta

Puede agregar 49 etiquetas más.

[Cancelar](#) [Crear](#)

5. Ahora el cluster se estará creando. Puede que tarde un rato, así que esperamos un poco
(Además da problemas, no me la deja crear a la primera, lo he hecho varias veces hasta que pude)
6. Una vez se cree, hacemos una **captura y la guardamos para la entrega**

Clústeres (1) Información

El clúster alba-digi-cluster se ha creado correctamente.

Clúster	Servicios	Tareas	Instancias de conte...	Supervisión de CloudWatch	Estrategia de proveedor de ca...
alba-digi-cluster	0	No hay tareas en ejecución.	0 EC2	Valor predeterminado	No se ha encontrado ningún v...

[Ver clúster](#) [X](#) [Última actualización: 20 de febrero de 2025, 13:30 \(UTC+1:00\)](#) [Clear cluster](#)

7. Clicamos en “view cluster”

Ya tenemos el cluster creado. Ahora vamos a por el siguiente paso

2. Crear un tipo de tarea para el servicio (task definition)

Ahora le tenemos que explicar al cluster que tarea va a ejecutar. Para eso la tenemos que definir. Recordemos que (por simplificar) tarea=contenedor. Vamos a definir esta tarea.

1. Viendo el cluster que acabáis de crear, a la izquierda tendréis una columna con diferentes opciones. Clicad en “task definition”

2. Le das a “Create new Task Definition”. Así definiremos la tarea

3. Clicais en “Fargate”. Hacemos esto porque queremos usar este tipo de instancias en lugar de las de EC2

4. Lo configuramos de la siguiente manera

- a. Task definition name: “alba-digi-task” -> Nombre de la tarea

Crear una nueva definición de tarea [Información](#)

Configuración de definición de tareas

Familia de definición de tareas [Información](#)

Especifique un nombre de familia de definición de tarea único.

Se permiten hasta 255 letras (mayúsculas y minúsculas), números, guiones y guiones bajos.

▼ Requisitos de infraestructura

Especifique los requisitos de infraestructura para la definición de tarea.

Tipo de lanzamiento [Información](#)

La selección del tipo de lanzamiento cambiará los parámetros de definición de la tarea.

 AWS Fargate

Computación sin servidor para contenedores.

 Instancias de Amazon EC2

Infraestructura autoadministrada mediante instancias de Amazon EC2.

Sistema operativo, arquitectura, modo de red

El modo de red se usa para las tareas y depende del tipo de computación seleccionado.

Sistema operativo/arquitectura [Información](#)

Modo de red [Información](#)

Tamaño de la tarea [Información](#)

Especifique la cantidad de CPU y memoria que reservar para su tarea.

CPU

Memoria

▼ Roles de tarea - *condicionales*

Rol de tarea [Información](#)

Un rol de IAM de tarea permite a los contenedores de la tarea realizar solicitudes de API a los servicios de AWS. Puede crear un rol de IAM de tarea desde la [consola de IAM](#).

- b. Task role: LabRole
- c. Network mode: awsvpc (la de por defecto) -> La red donde se conecta
- d. Operating system family: linux -> Si queremos linux o windows
- e. Task execution Role: LabRole -> El único rol que podemos poner con estas cuentas
- f. Task CPU: 1 cpu -> Cuanto cómputo le vamos a dar a esta tarea
- g. Task memory: 3GB -> Cuanta memoria máxima va a tener esta tarea

Especifique los requisitos de infraestructura para la definición de tarea.

Tipo de lanzamiento | Información
La selección del tipo de lanzamiento cambiará los parámetros de definición de la tarea.

AWS Fargate
Computación sin servidor para contenedores.

Instancias de Amazon EC2
Infraestructura autoadministrada mediante instancias de Amazon EC2.

Sistema operativo, arquitectura, modo de red
El modo de red se usa para las tareas y depende del tipo de computación seleccionado.

Sistema operativo/arquitectura | Información

Linux/X86_64 ▾

Modo de red | Información

awsvpc ▾

Tamaño de la tarea | Información
Especifique la cantidad de CPU y memoria que reservar para su tarea.

CPU

1 vCPU ▾

Memoria

3 GB ▾

▼ Roles de tarea - condicionales

Rol de tarea | Información
Un rol de IAM de tarea permite a los contenedores de la tarea realizar solicitudes de API a los servicios de AWS. Puede crear un rol de IAM de tarea desde la [consola de IAM](#).

LabRole ▾

Rol de ejecución de tareas | Información
El agente de contenedor utiliza un rol de IAM de ejecución de tareas para realizar solicitudes a la API de AWS en su nombre. Si aún no tiene un rol de IAM de ejecución de tareas creado, podi

LabRole ▾

▼ Ubicación de tareas - opcional

i Las restricciones de ubicación de tareas no son compatibles con el tipo de lanzamiento de AWS Fargate.

► Inyección de errores: opcional

Ahora llegaremos a una parte donde hay un botón que dice “Add container”. Si clicamos en él veremos que se abre una nueva ventana y que nos pide el nombre del contenedor, su url, etc. Aquí Amazon nos está pidiendo que le indiquemos qué contenedor va a ejecutar.

► Inyección de errores: opcional

▼ Contenedor: 1 Información

Detalles del contenedor
Especifique un nombre, una imagen de contenedor y si el contenedor debe marcarse como esencial. Cada definición de tarea debe tener al menos un contenedor esencial.

Nombre	URI de imagen	Contenedor esencial
wordpress	repository-url/image:tag	Sí ▾

Contenedor esencial [Eliminar](#)

Se permiten hasta 255 letras (mayúsculas y minúsculas), números, guiones y guiones bajos.

Registro privado | Información
Almacene las credenciales en Secrets Manager y, a continuación, utilice las credenciales para hacer referencia a las imágenes en los registros privados.

Autenticación de registro privado

Mapeos de puertos | Información
Agregue asignaciones de puertos para permitir que el contenedor acceda a los puertos del host para enviar o recibir tráfico. Para el nombre del puerto, se asignará un valor predeterminado si se deja en blanco.

Puerto del contenedor	Protocolo	Nombre del puerto	Protocolo de la aplicación	Eliminar
80	TCP	protocolo contenedor-puerto	HTTP	Eliminar

[Agregar mapeo de puertos](#)

Sistema de archivos raíz de solo lectura | Información
Cuando este parámetro está activado, el contenedor recibe acceso de solo lectura al sistema de archivos raíz.

Solo lectura

Límites de asignación de recursos - condicional | Información
Los límites de CPU, GPU y memoria del contenedor son diferentes de los valores de la tarea. Definen la cantidad de recursos que se asignan al contenedor. Si el contenedor intenta superar la memoria especificada en el límite estricto, el contenedor se cancela.

CPU	GPU	Límite estricto de memoria	Límite flexible de memoria
1	1	3 en GB	1 en GB

▼ Variables de entorno - opcional

Variables de entorno | Información

Para ello tenemos que subir nuestro contenedor a su repositorio de contenedores. Este servicio se llama ECR (Elastic Container Registry). Explicaros como hacer esto puede ser un poco liante y se nos va del objetivo de esta práctica, así que por ahora ignoraremos este proceso. Os vamos a dar directamente la URL del contenedor y toda esta información.

Por lo tanto clicamos en “Add Container” y añadimos lo siguiente: (no hay que clicar, viene por defecto)

- Container name: alba-digi-api-rest -> Nombre del contenedor
- Image: public.ecr.aws/n0h7v2z5/alba-digi-api-rest:latest -> URL del repositorio donde está el contenedor
- Memory Limits(Mib): (Soft Limit) 2 -> Cuanta memoria le vamos a dar al contenedor. Le vamos a dar la mayoría de la memoria que pusimos más arriba (antes hemos marcado 3GB)
- Port Mappings: 5050 - Protocol: TCP - App protocol: HTTP -> Significa que abrimos el puerto 5050, que es el que pide este contenedor según [su documentación](#)
- Environment:

- CPU Units: 1 (la que le hemos asignado a la tarea más arriba)

Container - 1 Essential container

Container details
Specify a name, container image, and whether the container should be marked as essential. Each task definition must have at least one essential container.

Name **Image URI** **Essential container**

Private registry **Info**
Store credentials in Secrets Manager, and then use the credentials to reference Images in private registries.

Port mappings **Info**
Add port mappings to allow the container to access ports on the host to send or receive traffic. For port name, a default will be assigned if left blank.

Container port	Protocol	Port name	App protocol
5050	TCP	container-port-protocol	HTTP

Add port mapping

Read only root file system **Info**
When this parameter is turned on, the container is given read-only access to its root file system.

Resource allocation limits - conditional **Info**
Container-level CPU, GPU, and memory limits are different from task-level values. They define how much resources are allocated for the container. If container attempts to exceed the memory specified in hard limit, the container is terminated.

CPU	GPU	Memory hard limit	Memory soft limit
1	1	3	2

In vCPU

Environment variables - optional

Environment variables **Info**
Add individually
Add a key-value pair to specify an environment variable.

Add environment variable

- El resto lo dejamos tal cual y le damos al botón que está abajo del todo que pone “Add”
(Este no está o no lo veo).

Docker labels - optional

Storage - optional

Ephemeral storage | Info

The amount of ephemeral storage, in GiB, to allocate for the task. By default, your tasks hosted on AWS Fargate receive a minimum of 20 GiB of ephemeral storage.

Amount

21

To specify a custom amount of ephemeral storage, specify a value between 21 GiB up to a maximum of 200 GiB.

Volumes | Info

Add one or more data volumes for your task to provide additional storage for the containers in the task. For each data volume, you must add a mount point to specify where to mount the data volume in the container.

Add volume

Volumes from | Info

Mount data volumes from another container.

Add volume from

Monitoring - optional

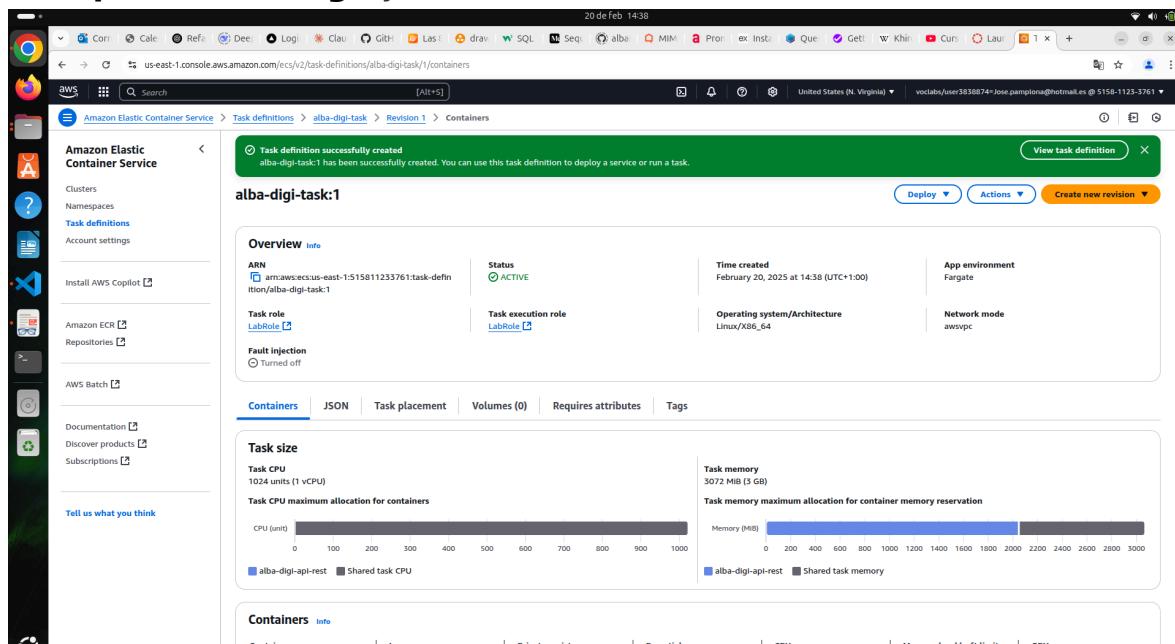
Configure your application trace and metric collection settings using the AWS Distro for OpenTelemetry integration.

Tags - optional | Info

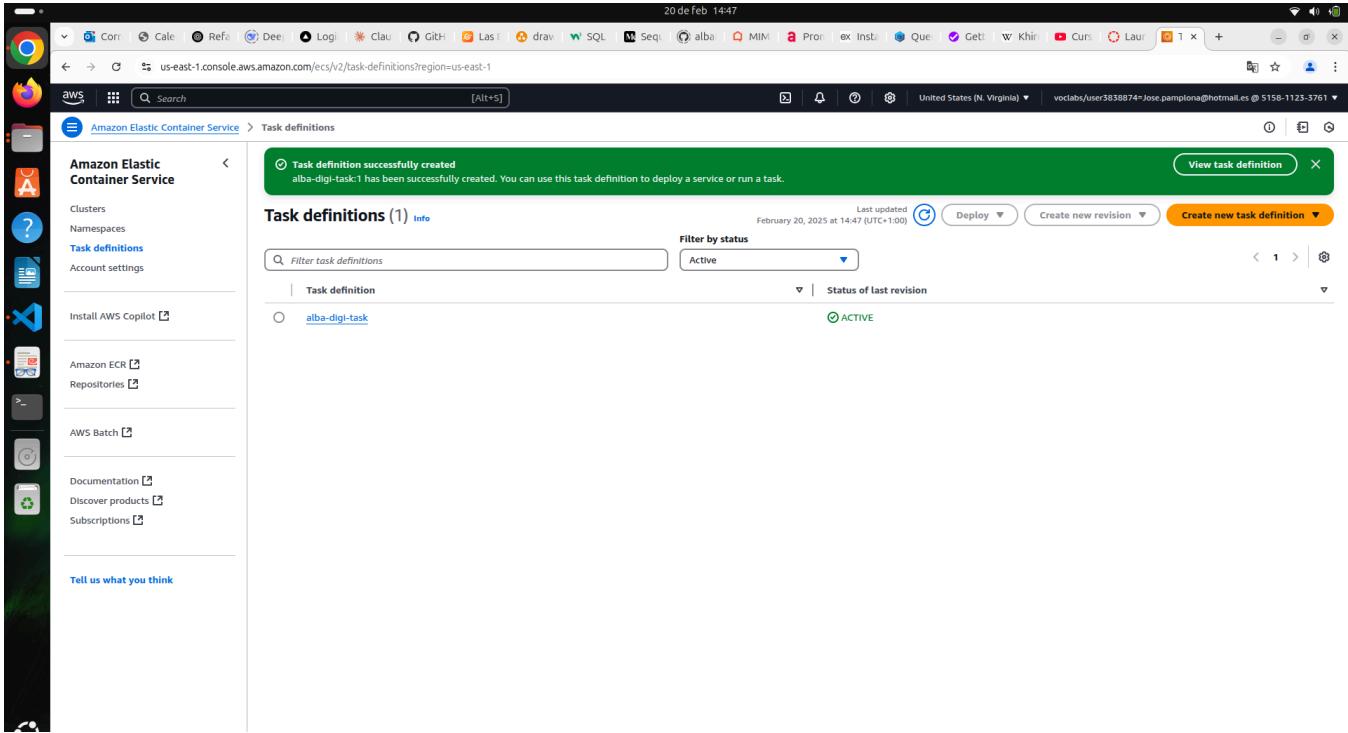
Tags help you to identify and organize your task definitions.

Create

- Y bajamos al final y le damos al botón que pone “Create”. Sacamos una foto para la entrega y vamos a “View Task Definition”



Así tendremos la task definition creada. Ahora podemos pasar al siguiente paso, crear el balanceador de carga



3. Crear un balanceador de carga

Ya tenemos un cluster (conjunto de máquinas) y una task definition (la tarea que va a hacer ese cluster). Pero ahora necesitamos alguien que nos indique a qué máquina del cluster va a llegar nuestra petición. Por eso necesitamos un balanceador de carga. Para ello vamos a hacer lo siguiente:

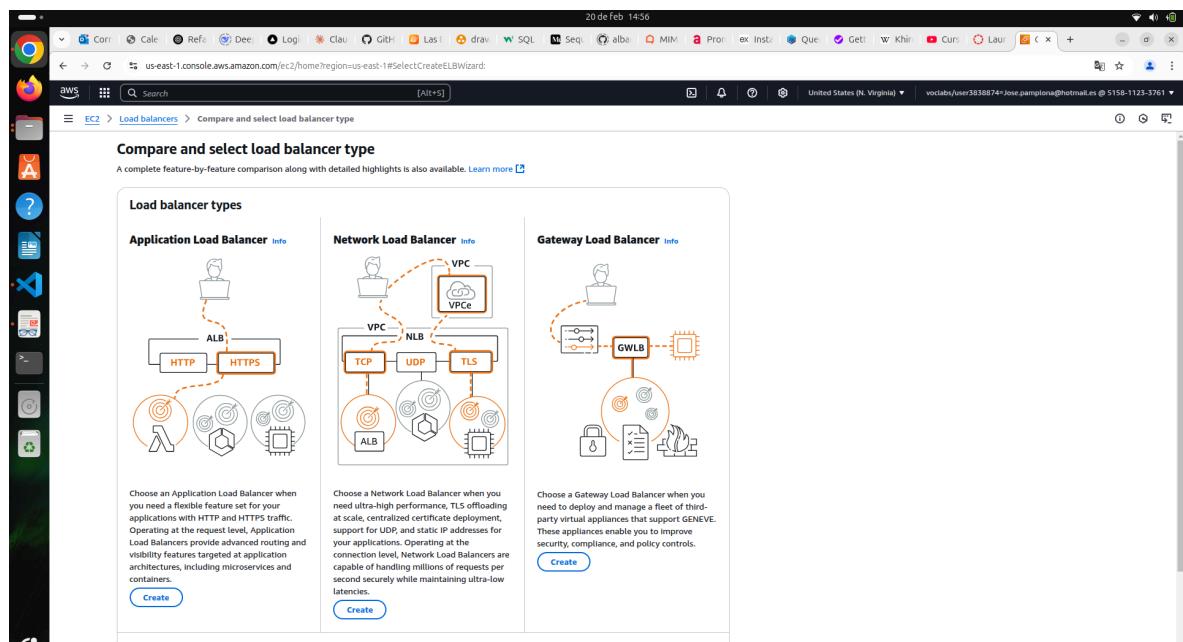
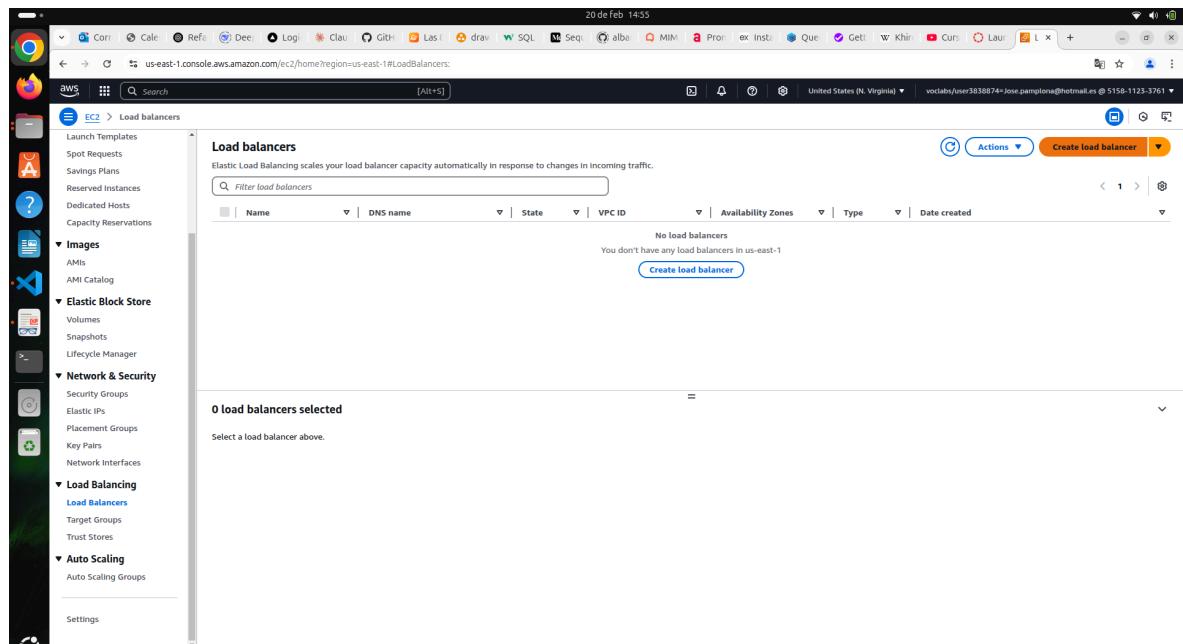
1. Vamos al servicio “EC2”

The screenshot shows the AWS EC2 task definitions console. On the left sidebar, under the 'Amazon EC2 Container' section, 'Task definitions' is selected. The main content area displays the EC2 service page with sections for 'Services', 'Features', and 'Resources'. The 'Services' section highlights EC2, EC2 Image Builder, and EC2 Global View. The 'Features' section includes EC2 Instances and AMIs. A modal window titled 'Introducing resource search' provides information about the new search feature.

- En la columna izquierda buscamos “Load Balancers” y clicamos en él.
Estará en la subsección “Load Balancing”

The screenshot shows the AWS EC2 home page. The left sidebar has 'Load Balancing' selected under the 'Network & Security' section. The main content area displays the 'Resources' section, which includes tables for Instances (running), Auto Scaling Groups, Capacity Reservations, Dedicated Hosts, Elastic IPs, Instances, Key pairs, Load balancers, Placement groups, Security groups, Snapshots, and Volumes. Below this, there are sections for 'Launch instance', 'Service health', 'Zones', 'Instance alarms', 'Scheduled events', and 'Explore AWS'.

- Clicamos en “Create Load Balancer” y luego en “Application Load Balancer”



4. Rellenamos la configuración así:
 - a. Load Balancer Name: alba-digi-load-balancer
 - b. Scheme: Internet-facing -> Queremos que se pueda acceder desde internet
 - c. IP address type: IPv4

Create Application Load Balancer

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

How Application Load Balancers work

Basic configuration

Load balancer name
Name must be unique within your AWS account and can't be changed after the load balancer is created.
alba-digi-balancer

Scheme
Scheme can't be changed after the load balancer is created.

- Internet-facing**
 - Serves Internet-facing traffic.
 - Has public IP addresses.
 - DNS name is publicly resolvable.
 - Requires a public subnet.
- Internal**
 - Serves internal traffic.
 - Has private IP addresses.
 - DNS name is publicly resolvable.
 - Compatible with the IPv4 and Dualstack IP address types.

Load balancer IP address type Info
Select the front-end IP address type to assign to the load balancer. The VPC and subnets mapped to this load balancer must include the selected IP address types. Public IPv4 addresses have an additional cost.

- IPv4**
Includes only IPv4 addresses.
- Dualstack**
Includes IPv4 and IPv6 addresses.
- Dualstack without public IPv4**
Includes a public IPv4 address, and private IPv4 and IPv6 addresses. Compatible with Internet-facing load balancers only.

Network mapping Info
The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC Info
The load balancer will exist and scale within the selected VPC. The selected VPC is also where the load balancer targets must be hosted unless routing to Lambda or on-premises targets, or if using VPC peering. To confirm the VPC for your targets, view target groups. For a new VPC, create a VPC.

d. En Network Mapping seleccionamos todas las zonas que haya y dejamos la subnet por defecto -> Queremos que pueda acceder a las redes por defecto de todas las zonas

Network mapping Info
The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC Info
The load balancer will exist and scale within the selected VPC. The selected VPC is also where the load balancer targets must be hosted unless routing to Lambda or on-premises targets, or if using VPC peering. To confirm the VPC for your targets, view target groups. For a new VPC, create a VPC.

Mappings Info
Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.

Availability Zones

- us-east-1a (us-east-1a)**
Subnet: subnet-062267ab16d9129f7 (IPv4 subnet CIDR: 172.31.0.0/16)
- us-east-1b (us-east-1b)**
Subnet: subnet-048e05793f1503ae (IPv4 subnet CIDR: 172.31.80.0/20)
- us-east-1c (us-east-1c)**
Subnet: subnet-046152e6ae7b54519 (IPv4 subnet CIDR: 172.31.16.0/20)
- us-east-1d (us-east-1d)**
Subnet: subnet-09fe6714330a066dc

e. Security groups dejamos el de por defecto -> Tema de seguridad sería otro capítulo.

The screenshot shows the configuration of a load balancer. Under 'Subnet', 'us-east-1f (use1-az5)' is selected. The 'IPv4 address' is listed as 'Assigned by AWS'. Under 'Security groups', 'default' is selected, with the identifier 'sg-09152bef72090a476' and VPC 'vpc-0df6b2ad81d7f898c'. A note states: 'A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can create a new security group.' A 'Listeners and routing' section is partially visible at the bottom.

5. Y ahora veremos la sección de Listeners and Routing.

The screenshot shows the 'Listeners and routing' section. It lists a single listener: 'Listener HTTP:80' with 'Protocol' set to 'HTTP' and 'Port' set to '80'. The 'Default action' is set to 'Forward to' with 'Select a target group' selected. A note says: 'A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.' Below the listener, there's a 'Listener tags - optional' section with a 'Create target group' button. Buttons for 'Add listener tag' and 'Add listener' are also present.

Ahora le vamos a explicar al balanceador a qué grupo de máquinas puede redirigir. Y para ello tenemos que crear ese grupo de máquinas. Será un grupo inteligente, así todo será automático. Será un “Target group”, así que vamos a clicar donde pone “Create target group”

The screenshot shows the 'Create target group' configuration. It has a 'Default action' section with 'Forward to' and 'Select a target group' selected. Below it is a 'Create target group' button with a blue icon.

Esto nos llevará a otra pantalla donde configuraremos el target group. Para ello rellenaremos la web con esta configuración:

1. Choose a target type: IP addresses (Las instancias de tipo Fargate son

targets de tipo IP por alguna razón, así que dejamos la configuración así)

2. Target Group Name: alba-digi-target-group

The screenshot shows the 'Specify group details' step of creating a target group. It's Step 1 of 2. The 'Basic configuration' section is visible, showing the 'Choose a target type' dropdown set to 'IP addresses'. Other options like 'Instances', 'Lambda function', and 'Application Load Balancer' are also shown with their respective descriptions. The 'Target group name' field contains 'alba-digi-target-group'. The 'Protocol : Port' section is present but not filled.

3. Protocol: HTTP

4. Port: 5050 -> Más arriba hemos abierto el puerto 5050, que es donde funciona este contenedor. Cuando el balanceador apunte a este grupo de instancias, queremos que apunte en el puerto 5050.

5. VPC: Seleccionamos la que hemos creado anteriormente junto al cluster (que debería ser la única)

6. Protocol Version: HTTP1

The screenshot shows the 'Protocol' step of creating a target group. The 'Protocol : Port' section has 'HTTP' selected in the dropdown and '5050' in the port field. Below it, the 'IP address type' section shows 'IPv4' selected. The 'VPC' section lists a single VPC entry: 'vpc-0df6b2ad81d7f898c IPv4 VPC CIDR: 172.31.0.0/16'. The 'Protocol version' section has 'HTTP1' selected. The 'Health checks' section at the bottom shows 'HTTP' selected as the health check protocol and '/>' as the path.

7. Ahora sí que vamos a configurar el healthcheck

a. Health Check Protocol: HTTP

b. Health Check Path: /alive

The screenshot shows the 'Create target group' configuration page in the AWS EC2 console. The 'Protocol version' section has 'HTTP1' selected. The 'Health checks' section shows 'Health check protocol' set to 'HTTP' and 'Health check path' set to '/alive'. There are sections for 'Attributes' and 'Tags - optional'. At the bottom right are 'Cancel' and 'Next' buttons.

8. Clicamos en Next. lo dejamos todo igual y luego en Create Target Group

The screenshot shows the 'Review targets' step of the 'Create target group' wizard. It lists the IP address '172.31.0' and port '5050'. A button 'Include as pending below' is visible. Below this, a table shows a single pending target with columns for Targets, Remove IPv4 address, Health status, IP address, Port, and Zone. The table header includes 'Targets (0)', 'Filter targets', 'Show only pending', and buttons for 'Remove all pending', 'Previous', and 'Create target group'.

The screenshot shows the AWS EC2 Target Groups console. A success message at the top indicates the target group 'alba-digi-target-group' was successfully created. The main pane displays the 'alba-digi-target-group' configuration, including its ARN, target type (IP), protocol (HTTP), port (5050), and VPC (vpc-0df6b2ad81d7f898c). Below this, a summary table shows 0 total targets, 0 healthy, 0 unhealthy, 0 unused, 0 initial, and 0 draining. The 'Targets' tab is active, showing a table with columns for IP address, Port, Zone, Health status, and Health status details. A note states 'No registered targets' and 'You have not registered targets to this group yet'. A 'Register targets' button is present.

Con esto ya tenemos nuestro target group. Ahora podemos seguir configurando nuestro balanceador de carga. Volvemos a la pestaña anterior, clicamos en el círculo más a la izquierda

The screenshot shows the AWS EC2 Target Groups console with the 'Target groups' list. A single target group, 'alba-digi-target-group', is listed with its details: ARN (arn:aws:elasticloadbalancing:us-east-1:515811233761:targetgroup/alba-digi-target-group/2313e15a79e1ebb9), Protocol (HTTP), Port (5050), Target type (IP), Load balancer (None associated), and VPC ID (vpc-0df6b2ad81d7f898c). The 'Actions' button is highlighted. The sidebar on the left shows various AWS services like Instances, Images, and Network & Security.

The screenshot shows the AWS EC2 Target groups page. At the top, there's a search bar and navigation links for 'EC2 > Target groups'. Below the header is a table with columns: Name, ARN, Port, Protocol, Target type, Load balancer, and VPC ID. A single row is listed: 'alba-digi-target-group' with ARN 'arn:aws:elasticloadbalancing...', Port '5050', Protocol 'HTTP', Target type 'IP', Load balancer 'None associated', and VPC ID 'vpc-0df6b2ad81d7f898c'. Below the table, it says '0 target groups selected' and 'Select a target group above.'

y en el desplegable seleccionamos el target group que acabamos de crear. Lo dejamos tal que así.

The screenshot shows the 'Listeners and routing' section for a load balancer. It displays a 'Listener HTTP:80' configuration. The 'Protocol' is set to 'HTTP' and 'Port' is '80'. The 'Default action' is 'Forward to' the 'alba-digi-target-group' target group, which is defined as 'Target type: IP, IPv4'. There are buttons for 'Remove', 'Edit', and 'Create target group'. Below this, there's a section for 'Listener tags - optional' with a button 'Add listener tag' and a note that you can add up to 50 more tags. At the bottom left is a button 'Add listener'.

Y le damos a 'Create Load Balancer'. Le damos a "view Load Balancer" y hacemos una **captura de pantalla para el entregable**. Ahora tenemos que esperar un poco hasta que en "State" cambie de "provisioning" a "Active".

EC2 > Load balancers > alba-digi-load-balancer

Successfully created load balancer: alba-digi-load-balancer
It might take a few minutes for your load balancer to fully set up and route traffic. Targets will also take a few minutes to complete the registration process and pass initial health checks.

alba-digi-load-balancer

Actions ▾

▼ Details

Load balancer type Application	Status Provisioning	VPC vpc-0df6b2ad81d7f890c	Load balancer IP address type IPv4
Scheme Internet-facing	Hosted zone Z355XD0TRQ7X7K	Availability Zones subnet-0d77cd1a920573138 us-east-1e (use1-a23) subnet-04e0a579a3f1383ae us-east-1b (use1-a22) subnet-062267ab16d9129f7 us-east-1a (use1-a21) subnet-046132e8ae7b54319 us-east-1c (use1-a24) subnet-09fe6714530a066dc us-east-1d (use1-a26) subnet-04a9ee57483dcfc7b us-east-1f (use1-a25)	Date created February 22, 2025, 18:16 (UTC+01:00)
Load balancer ARN arn:aws:elasticloadbalancing:us-east-1:515811235761:loadbalancer/app/alba-digi-load-balancer/469af35249134362	DNS name info alba-digi-load-balancer-1171243710.us-east-1.elb.amazonaws.com (A Record)		

Listeners and rules | Network mapping | Resource map | Security | Monitoring | Integrations | Attributes | Capacity | Tags

Listeners and rules (1) Info

A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.

Manage rules ▾ | **Manage listener ▾** | **Add listener**

Filter listeners

EC2 > Load balancers

Load balancers (1/1)

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

Filter load balancers

Name	DNS name	State	VPC ID	Availability Zones	Type	Date created
alba-digi-load-balancer	alba-digi-load-balancer-117...	Active	vpc-0df6b2ad81d7f898c	6 Availability Zones	application	February 22, 2025, 18:16 (UTC+01:00)

Load balancer: alba-digi-load-balancer

Details Listeners and rules Network mapping Resource map Security Monitoring Integrations Attributes Capacity Tags

Details

Load balancer type	Status	VPC	Load balancer IP address type
Application	Active	vpc-0df6b2ad81d7f898c	IPv4
Scheme	Hosted zone	Availability Zones	Date created
Internet-facing	Z355XKDTRQ7X7K	subnet-0d77cd1a920573158 (us-east-1e (use1-az3)) subnet-048ea579a3f1383ae (us-east-1b (use1-az2)) subnet-062267ab16d9129f7 (us-east-1a (use1-az1)) subnet-046132e8ae7b54319 (us-east-1c (use1-az4)) subnet-09fe6714530a66dc (us-east-1d (use1-az5))	February 22, 2025, 18:16 (UTC+01:00)

Ya tenemos nuestro balanceador de carga, por lo que podemos empezar con nuestro último paso.

4. Crear el servicio

Ahora tenemos un cluster (conjunto de máquinas), una task definition (una tarea tipo) y un balanceador de carga (quien sabe a donde apuntar). Ahora necesitamos juntar todo esto, y para ello vamos a crear un servicio. Para ello haremos lo

siguiente:

1. Volveremos al cluster que hemos creado anteriormente. Para ello vamos al servicio “Elastic Container Service” y clicamos en él: alba-digi-cluster

The screenshot shows the AWS ECS console interface. On the left, there's a sidebar with navigation links like 'Clusters', 'Namespaces', 'Task definitions', and 'Account settings'. The main area has a title 'Clusters (1) info' and a table with one row for 'alba-digi-cluster'. The table columns include Cluster, Services, Tasks, Container instances, CloudWatch monitoring, and Capacity provider strategy. Below the table, there's a 'Create cluster' button. To the right of the table, there's a detailed view of the cluster 'alba-digi-cluster'. This view includes sections for 'Services' (listing 'Elastic Container Service' and 'Elastic Container Registry'), 'Features' (listing 'Contacts' and 'Container services'), 'Resources' (with a note about resource search), 'Documentation' (with a link to 'AWS Elastic Container Service'), and 'Actions' (with a 'Create load balancer' button). At the bottom of the detailed view, there's a list of IP addresses associated with the cluster.

Cluster	Services	Tasks	Container instances	CloudWatch monitoring	Capacity provider strategy
alba-digi-cluster	0	No tasks running	0 EC2	Default	No default found

Load balancer IP address type: IPv4
Date created: February 22, 2025, 18:16 (UTC+01:00)

IP Address	Description
3205751130	us-east-1e (use1-az3)
a3f1303ae	us-east-1b (use1-az2)
16d9129f7	us-east-1a (use1-az1)
ue7054319	us-east-1c (use1-az4)
330066dc6	us-east-1d (use1-az6)
subnet-0449657483dcc77b	us-east-1f (use1-az5)

2. Abajo veremos que está seleccionada la pestaña “Services”. Clicaremos en el botón de abajo, donde pone “Create”. Así empezaremos a crear el servicio

The screenshot shows the 'Cluster overview' section of the ECS console. It displays the ARN of the cluster, its status as 'Active', CloudWatch monitoring settings, and task metrics. Below this, there's a table for tasks showing one pending task and one running task. A navigation bar at the bottom includes tabs for Services, Tasks, Infrastructure, Metrics, Scheduled tasks, Configuration, and Tags. A prominent 'Create' button is located in the center of the main content area.

3. Lo configuraremos tal que así:

- Compute options: Launch type
- Launch Type: Fargate (Habíamos decidido no usar ec2, y usar fargate)
- Platform version: latest
- Application type: Service

This screenshot shows the 'Create' service configuration page. In the 'Environment' section, 'alba-digi-cluster' is selected. Under 'Compute configuration (advanced)', the 'Launch type' is set to 'FARGATE'. In the 'Deployment configuration' section, the 'Application type' is set to 'Service'. The 'Task definition' dropdown is visible at the bottom.

- Task definition - Family: alba-digi-task (la creada anteriormente)
- Task Definition - Revision: La que ponga (latest)
- Service Name: alba-digi-service
- Number of tasks: 2 (vamos a poner más de una, que ya que creamos el servicio...)
- Deployment options: rolling update, min 100%, max 200%

Amazon Elastic Container Service

Clusters

- Namespaces
- Task definitions
- Account settings

Install AWS Copilot

Amazon ECR

Repositories

AWS Batch

Documentation

Discover products

Subscriptions

Tell us what you think

Family alba-digi-task **Revision** 2 (LATEST)

Service name alba-digi-service

Service type Replica

Desired tasks 2

Availability Zone rebalancing Turn on Availability Zone rebalancing

Deployment options

Deployment type Rolling update

Min running tasks % 100

Max running tasks % 200

j. Deployment failure protection: desmarcamos la casilla de circuit breaker, ya que no queremos que actúe.

Deployment options

Deployment type Rolling update

Min running tasks % 100

Max running tasks % 200

Deployment failure detection

⚠ Deployment protection settings have been turned off.

Use the Amazon ECS deployment circuit breaker

Use CloudWatch alarm(s)

4. En la misma página, en el apartado 'Load Balancing':
 - a. Load Balancer Type: Application Load Balancer
 - b. Use existing load balancer
 - c. Seleccionamos el que hemos creado en el paso anterior
 - d. Donde pone Health Check grace period ponemos 30 (le vamos a dar 30 segundos antes de empezar a verificar si el servicio está sano)
 - e. Load balancer name: alba-digi-load-balancer
 - f. En Container to load balance: Seleccionamos alba-digi-api-rest:5050:5050
 - g. Use an existing listener -> 80:HTTP

Amazon Elastic Container Service

Clusters

- Namespaces
- Task definitions
- Account settings

Install AWS Copilot

Amazon ECR

Repositories

AWS Batch

Documentation

Discover products

Subscriptions

Tell us what you think

▼ Load balancing - optional

Configure load balancing using Amazon Elastic Load Balancing to distribute traffic evenly across the healthy tasks in your service.

Use load balancing

Load balancer type | Info

Specify the load balancer type to distribute incoming traffic across the tasks running in your service.

Application Load Balancer
An Application Load Balancer makes routing decisions at the application layer (HTTP/HTTPS), supports path-based routing, and can route requests to one or more ports.

Network Load Balancer
A Network Load Balancer makes routing decisions at the transport layer (TCP/UDP).

Container

The container and port to load balance the incoming traffic to

alba-digi-api-rest 5050:5050

Host port:Container port

Application Load Balancer

Specify whether to create a new load balancer or choose an existing one.

Create a new load balancer

Use an existing load balancer

Load balancer

Select the load balancer you wish to use to distribute incoming traffic across the tasks running in your service.

alba-digi-load-balancer

Health check grace period | Info

30

seconds

Listener | Info

Specify the port and protocol that the load balancer will listen for connection requests on.

Create new listener

Use an existing listener

Listener

80:HTTP

Listener rules for 80:HTTP (1)

Traffic received by the listener is routed according to its rules. Rules are evaluated in priority order, from the lowest value to the highest value. The default rule is evaluated last.

Evaluation order	Rule path	Target group
default	/	alba-digi-target-group

h. Use an existing target group: alba-digi-target-group - /alive - HTTP

5. Hacemos un pantallazo a toda la pantalla (que se vean todos los campos) y lo añadimos a nuestra entrega. Luego le damos a “Create”.

Amazon Elastic Container Service

Clusters

- Namespaces
- Task definitions
- Account settings

Install AWS Copilot

Amazon ECR

Repositories

AWS Batch

Documentation

Discover products

Subscriptions

Tell us what you think

Load balancer

Select the load balancer you wish to use to distribute incoming traffic across the tasks running in your service.

alba-digi-load-balancer

Health check grace period | Info

30

seconds

Listener | Info

Specify the port and protocol that the load balancer will listen for connection requests on.

Create new listener

Use an existing listener

Listener

80:HTTP

Listener rules for 80:HTTP (1)

Traffic received by the listener is routed according to its rules. Rules are evaluated in priority order, from the lowest value to the highest value. The default rule is evaluated last.

Evaluation order	Rule path	Target group
default	/	alba-digi-target-group

Target group | Info

Specify whether to create a new target group or choose an existing one that the load balancer will use to route requests to the tasks in your service.

Create new target group

Use an existing target group

Target group name

alba-digi-target-group

Health check path

/alive

Health check protocol

HTTP

Evaluation order ▾ Rule path ▾ Target group ▾

default / alba-digi-target-group

Target group Info Specify whether to create a new target group or choose an existing one that the load balancer will use to route requests to the tasks in your service.

Create new target group
 Use an existing target group

Target group name alba-digi-target-group

Health check path /alive

Health check protocol HTTP

VPC Lattice - optional Info Fully managed application networking service to connect, secure, and monitor your services across multiple accounts and virtual private clouds (VPCs). When you use VPC Lattice, there is a cost associated with it.

Service auto scaling - optional Automatically adjust your service's desired count up and down within a specified range in response to CloudWatch alarms. You can modify your service auto scaling configuration at any time to meet the needs of your application.

Volume - optional Info Configure a data volume to provide additional storage for the containers in the task.

Tags - optional Info Tags help you to identify and organize your resources.

[Cancel](#) [Create](#)

6. Volvemos a la pantalla del clúster, y **hacemos un pantallazo que se vea el servicio**. Después, clicamos en nuestro servicio para ver cómo está en su interior.

us-east-1.console.aws.amazon.com/ecs/v2/clusters/alba-digi-cluster/services?region=us-east-1

Amazon Elastic Container Service > Clusters > alba-digi-cluster > Services

alba-digi-cluster

Cluster overview

ARN	Status	CloudWatch monitoring	Registered container instances
arn:aws:ecs:us-east-1:515811235761:cluster/alba-digi-cluster	Active	Default	-

Services

Draining	Status	Tasks	Running
-	Active	Pending	2

Services (1)

Filter services by value	Filter launch type	Filter service type
<input type="checkbox"/> Service name	Any launch type	Any service type
<input type="checkbox"/> alba-digi-service	arn:aws:ecs:us-east-1:515811235761:service/alba-digi-service	REPLICA
	2/2 Tasks running	Completed
		alba-digi-task-2

The screenshot shows the AWS ECS console for the 'alba-digi-cluster'. On the left, there's a sidebar with various AWS services like Amazon ECR, AWS Batch, and Documentation. The main area is titled 'alba-digi-service' and shows the 'Info' tab. Under 'Service overview', it says 'Status: Active' with 2 Desired tasks, 0 Pending, and 2 Running. The 'Task definition: revision alba-digi-task:2' is listed. The 'Deployment status' is 'Success'. The 'Health and metrics' tab is selected, showing 'Health check grace period: 30 seconds'. Below that, there's a table for 'Load balancer health' with one entry: 'alba-digi-load-balancer' (Application Load Balancer) with port 'HTTP:80' and target group 'alba-digi-target-group'. The target group has 2 healthy and 0 unhealthy targets. At the bottom, there are sections for 'Health' and 'CPU utilization' (59.6%) and 'Memory utilization' (8.43%).

¡Ya tenemos nuestro servicio funcionando! Vamos a probarlo y podemos dar la práctica por terminada.

5. Probar el servicio

Para poder dar por buena la práctica tenéis que añadir los **siguientes 3 pantallazos** (los más importantes!)

1. Un pantallazo del servicio con sus 2 tasks en running, tal que así.

The screenshot shows the 'Tasks' page for the 'alba-digi-service'. The top navigation bar includes 'Amazon Elastic Container Service > Clusters > alba-digi-cluster > Services > alba-digi-service > Tasks'. Below the navigation, there are tabs for 'Health and metrics', 'Tasks' (which is selected), 'Logs', 'Deployments', 'Events', 'Configuration and networking', and 'Tags'. The main area is titled 'Tasks (1/2)' and shows two tasks. Each task row includes a checkbox, a task ID (32368... and 85a46...), the status 'Running', the container name 'alba-digi...', and the health status 'Unknown'. There are also columns for 'Last status', 'Desired status', 'Ta...', 'Health sta...', 'Started at', 'Container instan...', and 'La...'. At the bottom, there are buttons for 'C' (Create), 'Update service', and 'Delete service'. Below the table, there are filters: 'Filter tasks by property or value', 'Filter desired status' (set to 'Running'), and 'Filter launch type' (set to 'Any launch type').

en el caso de la práctica mia, sería:

Amazon Elastic Container Service > Clusters > alba-digi-cluster > Services > alba-digi-service > Tasks

alba-digi-service Info

Service overview

Status: Active | Tasks (2 Desired) | Task definition: revision alba-digi-task-2 | Deployment status: Success

Health and metrics | **Tasks** | Logs | Deployments | Events | Configuration and networking | Service auto scaling | Tags

Tasks (1/2)

Task	Last status	Desired state	Task def.	Health state	Started by	Started at	Container instance	Launch type	Platform ...
a2330...	Running	Running	alba-digi...	Unknown	ecs-svc/09983212138...	11 minutes ago	-	FARGATE	1.4.0
da2e0...	Running	Running	alba-digi...	Unknown	ecs-svc/09983212138...	11 minutes ago	-	FARGATE	1.4.0

Containers for task a23308f01d55405fb9180fabf2e305bc

Containers (1)							
Container name	Container runtime ID	Image URI	Image Digest	Status	Health status	CPU	Memory hard/soft limit
alba-digi-api-rest	a23308f01d55405...	public.ecr...	sha256:f9...	Running	Unknown	0	- / 2 GB

2. Un **pantallazo de los eventos del servicio**, donde se vea que el servicio está steady (puede que tarde un poco en aparecer)

Amazon Elastic Container Service > Clusters > alba-digi-cluster > Services > alba-digi-service > Events

alba-digi-service Info

Health and metrics | Tasks | Logs | Deployments | **Events** | Configuration and networking | Tags

Events (4)

Started at	Message	Event ID
January 04, 2024 at 18:37 (UTC+1:00)	service alba-digi-service has reached a steady state.	d65112f4-e90c-4041-38036f7231c7
January 04, 2024 at 18:37 (UTC+1:00)	service alba-digi-service deployment ecs-svc/5797995502118074585 deployment completed.	91f13654-18e4-42cd-bb57a4328345
January 04, 2024 at 18:37 (UTC+1:00)	service alba-digi-service registered 2 targets in target-group alba-digi-target-group	2dbf5385-6973-4458-025e0b094c91
January 04, 2024 at 18:37 (UTC+1:00)	service alba-digi-service has started 2 tasks: task 323682339da246c29934b69bbf54fd99 task 85a4612b00b6445fa667dc92402d3945.	ae99ea64-96ad-42b3-1476074d1d41

en el caso de la práctica mia, sería:

3. Una prueba de que el servicio está ejecutándose. Para ello vamos a hacer lo siguiente:
 - a. Ir al servicio EC2
 - b. Ir al apartado “Load Balancers”
 - c. Copiar el dns name, lo podeis encontrar aquí

	Name	DNS name
	alba-digi-load-balancer	alba-digi-load-balancer-956276442.eu-central-1.elb.amazonaws.com

en el caso de la práctica mia, sería:

Load balancers (1)									
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.									
Filter load balancers									
Name	DNS name	State	VPC ID	Availability Zones	Type	Date created			
alba-digi-load-balancer	alba-digi-load-balancer-1171243710.us-east-1.elb.amazonaws.com	Active	vpc-0df6b2ad81d7fb98c	6 Availability Zones	application	February 22, 2025, 1			

- d. Abrir una terminal y escribir “curl -v http://{SUSTITUYE}/alive” donde la palabra SUSTITUYE es el DNS Name de arriba. Sería algo así

```
$ curl -v http://alba-digi-load-balancer-624687030.eu-central-1.elb.amazonaws.com/alive
```

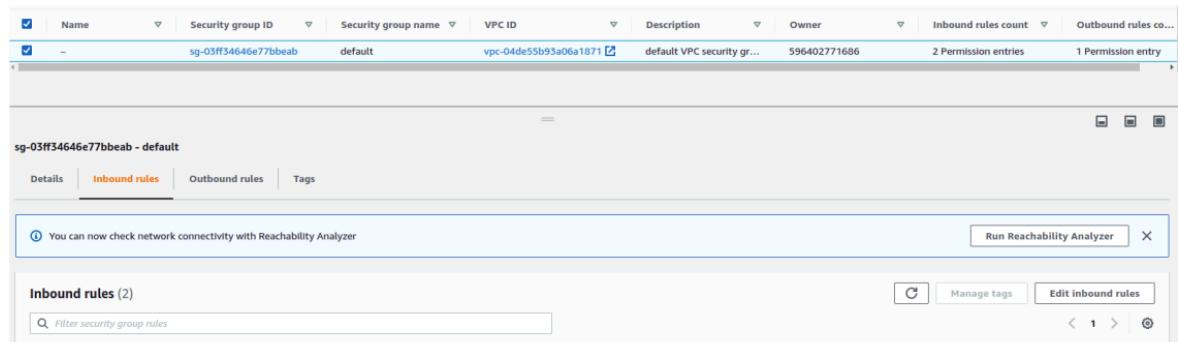
en el caso de la práctica mia, sería:

```
curso@koldo:~/workspace/curso/amazon-web-services/tema4/practicas/Koldo_Sanmartin_AWS_practica_4$ curl -v http://alba-digi-load-balancer-1171243710.us-east-1.elb.amazonaws.com/alive
```

quedando:

```
curso@koldo:~/workspace/curso/amazon-web-services/tema4/practicas/Koldo_Sanmartin_AWS_practica_4$ curl -v http://alba-digi-load-balancer-1171243710.us-east-1.elb.amazonaws.com/alive
* Host alba-digi-load-balancer-1171243710.us-east-1.elb.amazonaws.com:80 was resolved.
* IPv6: (none)
* IPv4: 44.221.243.87, 23.23.60.73
* Trying 44.221.243.87:80...
* connect to 44.221.243.87 port 80 from 192.168.1.137 port 37716 failed: Expiró el tiempo de conexión
* Trying 23.23.60.73:80...
* connect to 23.23.60.73 port 80 from 192.168.1.137 port 41168 failed: Expiró el tiempo de conexión
* Failed to connect to alba-digi-load-balancer-1171243710.us-east-1.elb.amazonaws.com port 80 after 268083 ms: Couldn't connect to server
* Closing connection
curl: (28) Failed to connect to alba-digi-load-balancer-1171243710.us-east-1.elb.amazonaws.com port 80 after 268083 ms: Couldn't connect to server
curso@koldo:~/workspace/curso/amazon-web-services/tema4/practicas/Koldo_Sanmartin_AWS_practica_4$
```

e. Como veis falla. Es porque al crear una VPC tenéis que abrir esa red a vuestro internet. Para ello vais al servicio EC2, Security groups, seleccionamos el que pone “default” y le dais a Inbound rules, edit Inbound rules



en mi caso:

Security Groups (3) <small>Info</small>						
<input type="text" value="Find resources by attribute or tag"/> Actions Export security groups to CSV Create security group						
	Name	Security group ID	Security group name	VPC ID	Description	Owner
<input type="checkbox"/>	-	sg-02060f76e9b2f97cb	default	vpc-04dcba998e1a08c5	default VPC security group	515811233761
<input type="checkbox"/>	-	sg-09152bef72090a476	default	vpc-04dcba998e1a08c5	default VPC security group	515811233761
<input type="checkbox"/>	-	sg-07e9df215dab0bd84	launch-wizard-1	vpc-04dcba998e1a08c5	launch-wizard-1 created 2025-02-19T1...	515811233761

Security Groups (3) <small>Info</small>						
<input type="text" value="Find resources by attribute or tag"/> Actions Export security groups to CSV Create security group						
	Name	Security group ID	Security group name	VPC ID	Description	Owner
<input type="checkbox"/>	-	sg-02060f76e9b2f97cb	default	vpc-04dcba998e1a08c5	default VPC security group	515811233761
<input type="checkbox"/>	-	sg-09152bef72090a476	default	vpc-04dcba998e1a08c5	default VPC security group	515811233761
<input type="checkbox"/>	-	sg-07e9df215dab0bd84	launch-wizard-1	vpc-04dcba998e1a08c5	launch-wizard-1 created 2025-02-19T1...	515811233761

sg-02060f76e9b2f97cb - default																																		
Details Inbound rules Outbound rules Sharing - new VPC associations - new Tags																																		
Inbound rules (1) Manage tags Edit inbound rules																																		
<table border="1"> <thead> <tr> <th colspan="7">Inbound rules (1)</th> </tr> <tr> <th colspan="7"> <input type="text" value="Search"/> Actions </th> </tr> <tr> <th></th> <th>Name</th> <th>Security group rule ID</th> <th>IP version</th> <th>Type</th> <th>Protocol</th> <th>Port range</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/></td> <td>-</td> <td>sgr-0b20ac9b2095715c5</td> <td>-</td> <td>All traffic</td> <td>All</td> <td>All</td> </tr> </tbody> </table>							Inbound rules (1)							<input type="text" value="Search"/> Actions								Name	Security group rule ID	IP version	Type	Protocol	Port range	<input type="checkbox"/>	-	sgr-0b20ac9b2095715c5	-	All traffic	All	All
Inbound rules (1)																																		
<input type="text" value="Search"/> Actions																																		
	Name	Security group rule ID	IP version	Type	Protocol	Port range																												
<input type="checkbox"/>	-	sgr-0b20ac9b2095715c5	-	All traffic	All	All																												

f. Le dais a add rule y lo dejáis así

sg-0d63e09eeaa0a2947a	All traffic	All	All	My IP	<input type="text"/>	Delete
-----------------------	-----------------------------	---------------------	---------------------	-----------------------	----------------------	------------------------

g. Le dais a save rule y volveis a probar a hacer la petición en la terminal. Os debería salir lo siguiente

```
jbermejo@jbermejo-ThinkPad-L14-Gen-1:~$ curl -v http://alba-digi-load-balancer-624687030.eu-central-1.elb.amazonaws.com/alive
*   Trying 3.74.107.125:80...
* TCP_NODELAY set
* Connected to alba-digi-load-balancer-624687030.eu-central-1.elb.amazonaws.com (3.74.107.125) port 80 (#0)
> GET /alive HTTP/1.1
> Host: alba-digi-load-balancer-624687030.eu-central-1.elb.amazonaws.com
> User-Agent: curl/7.68.0
> Accept: /*

* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Date: Tue, 17 Jan 2023 18:12:08 GMT
< Content-Type: text/plain; charset=UTF-8
< Content-Length: 76
< Connection: keep-alive
<
* Connection #0 to host alba-digi-load-balancer-624687030.eu-central-1.elb.amazonaws.com left intact
/ - Hello alive! Host:ip-10-0-1-149.eu-central-1.compute.internal/10.0.1.149jbermejo@jbermejo-ThinkPad-L14-Gen-1:~$
```

Que entregar

Recordemos que para terminar esta entrega hay que entregar:

1. Foto de la creación del cluster
2. Foto de la creación de la task definition
3. Foto de la creación del balanceador de carga
4. Foto de la creación del servicio
5. Foto de la terminal donde se ver que el servicio funciona

Conclusiones

En esta práctica hemos levantado un servicio con múltiples máquinas. Hemos dejado de lado muchísimos temas de seguridad y buenas prácticas porque no nos daba tiempo. Os animamos a que investigueis estos temas, ya que son imprescindibles si queremos implantar estas tecnologías en una empresa. Recordad borrar todo y si tenéis alguna duda preguntadnos sin miedo!