

# Manual despliegue API y TABLERO Flight\_Delay

## 1. Requisitos de Infraestructura en AWS

- Aprovechando la infraestructura AWS se lanza una instancia; recomendable una parametrización con las siguientes recomendaciones: Ubuntu como sistema operativo, instancia t2-medium, y una capacidad mínima de 20 GB.
- Asegúrate de configurar la apertura de puertos necesarios en las reglas de entrada del grupo de seguridad que vas a asociar a la instancia EC2 y más adelante al clúster ECS. En este caso, se utilizan los puertos 8001 y 8050 para la API y el Tablero respectivamente.

**Resumen de instancia de i-05899dfc80712b154 (contenedores)**

Se ha actualizado hace less than a minute

ID de la instancia: i-05899dfc80712b154 (contenedores)

Dirección IPv6: -

Tipo de nombre de anfitrión: -

Nombre de IP: ip-172-31-81-62.ec2.internal

Responder al nombre DNS de recurso privado IPv4 (A): -

Dirección IP asignada automáticamente: 54.146.160.162 [IP pública]

Rol de IAM: -

IMDSv2: Required

Dirección IPv4 pública: 54.146.160.162 [dirección abierta]

Estado de la instancia: En ejecución

Nombre DNS de IP privada (solo IPv4): ip-172-31-81-62.ec2.internal

Tipo de instancia: t2.medium

ID de VPC: vpc-082afa25e638376e9

ID de subred: subnet-0c76144f3dffa63f0

Direcciones IPv4 privadas: 172.31.81.62

DNS de IPv4 pública: ec2-54-146-160-162.compute-1.amazonaws.com [dirección abierta]

Direcciones IP elásticas: -

Hallazgo de AWS Compute Optimizer: Suscríbete a AWS Compute Optimizer para recibir recomendaciones. [Más información]

Nombre del grupo de Auto Scaling: -

**Detalles de la instancia**

Plataforma: Ubuntu (inferido)

Detalles de la plataforma: Linux/UNIX

Detener la protección: desactivado

ID de AMI: ami-0fc5d935ebf8bc3bc

Nombre de AMI: ubuntu/images/hvm-ssd/ubuntu-jammy-22.04-amd64-server-20230919

Hora de lanzamiento: Tue Nov 21 2023 18:48:24 GMT-0500 (hora estándar de Colombia) (less than a minute)

Monitoreo: desactivado

Protección de terminación: desactivado

Ubicación de AMI: amazon/ubuntu/images/hvm-ssd/ubuntu-jammy-22.04-amd64-server-20230919

**Reglas de entrada**

Nombre	ID de la regla del grupo d...	Intervalo de pu...	Protocolo	Origen	Grupos de seguridad	Descripción
-	sgr-0e7c2132a68b292d5	8001	TCP	0.0.0.0/0	launch-wizard-1	-
-	sgr-0df2bec0196ccb107	8050	TCP	0.0.0.0/0	launch-wizard-1	-
-	sgr-024d0da3c3d556b59	22	TCP	0.0.0.0/0	launch-wizard-1	-

## 2. Preparación de la máquina EC2 y Docker:

Es necesario realizar las configuraciones y actualizaciones en la máquina virtual (EC2) de los paquetes requeridos incluyendo python, pip, Docker, dependencias e instaladores requeridos; las sugerencias incluyen la realización de los siguientes pasos:

- Conéctate a la instancia utilizando el siguiente comando  
`ssh -i llave .pem ubuntu@IP`
- Ahora pasamos a instalar Docker en la máquina, para lo cual requerimos eliminar posibles versiones anteriores y agregar el repositorio de la última versión estable de Docker.  
`sudo apt -get remove docker docker - engine docker .io containerd runc`
- Actualice el índice de paquetes  
`sudo apt -get update`

- Instale dependencias para verificar certificados (ca-certificates), obtener objetos con su URL (curl) y administrar llaves PGP (gnupg)

`sudo apt -get install ca - certificates curl gnupg`

- Agregue la llave de Docker

`sudo install -m 0755 -d /etc /apt/ keyrings`

`curl -fsSL https :// download . docker . com/ linux / ubuntu /gpg | sudo gpg -- dearmor - o /etc /apt/ keyrings / docker . gpg`

`sudo chmod a+r / etc/apt/ keyrings / docker . gpg`

- Agregue el repositorio de Docker a su sistema para la instalación

`echo \`

`"deb [ arch ="$( dpkg --print - architecture )" signed -by =/ etc/apt/ keyrings / docker . gpg] https :// download . docker . com/ linux / ubuntu \`

`"$( ./etc/os - release && echo " $VERSION_CODENAME )" stable " | \`

`sudo tee /etc /apt/ sources . list .d/ docker . list > /dev / null`

- Actualice nuevamente el índice de paquetes con este nuevo repositorio incluido

`sudo apt -get update`

- Instale Docker Engine, containerd, y Docker Compose

`sudo apt -get install docker -ce docker -ce -cli containerd .io docker -buildx -`

`plugin docker - compose - plugin`

- Para verificar su instalación, descargue, construya y ejecute la imagen hello-world

`sudo docker run hello – world`

- Verificamos que se encuentre la imagen con el comando

`sudo docker images`

```
ubuntu@ip-172-31-81-62:~$ sudo docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
hello-world         latest         9c7a54a9a43c   6 months ago   13.3kB
ubuntu@ip-172-31-81-62:~$
```

- Clonamos el repositorio en el que se encuentran los archivos necesarios para el correcto despliegue:

`git clone https://github.com/dhoyoso/proyecto_despliegue_soluciones.git`

```
ubuntu@ip-172-31-81-62:~$ git clone https://github.com/dhoyoso/proyecto_despliegue_soluciones.git
Cloning into 'proyecto_despliegue_soluciones'...
remote: Enumerating objects: 348, done.
remote: Counting objects: 100% (40/40), done.
remote: Compressing objects: 100% (31/31), done.
remote: Total 348 (delta 17), reused 23 (delta 9), pack-reused 308
Receiving objects: 100% (348/348), 54.71 MiB | 22.46 MiB/s, done.
Resolving deltas: 100% (138/138), done.
```

**NOTA:** En el ambiente de desarrollo se realizaron las pruebas en el paquete del modelo con Tox para lo cual se crearon dos ambientes train y test\_package y que se encuentra dentro del repositorio; teniendo en cuenta que la API y el Tablero han superado las validaciones respectivas se invita a desplegar sin la ejecución de los comandos de pruebas; se dejan a manera para su exploración voluntaria.

✓ Instale tox, una librería de automatización y pruebas

```

    pip install tox
    sudo apt -get install tox
✓ Incluye la dirección /home/ubuntu/.local/bin en el PATH para facilitar su ejecución
    PATH = $PATH :/ home / ubuntu /. local / bin
✓ Ingrese a la carpeta model-package y ejecute el ambiente de prueba del api de tox;
    cd model-package
    tox run -e train
✓ Verifique que se pasen todas las pruebas.
✓ Ejecute ahora el ambiente de ejecución de tox
    tox run -e test_package

```

(Los comandos presentados se usan en el paquete del modelo, sin embargo, el API también cuenta con empaquetamiento con TOX así que puedes repetir estos pasos con la carpeta que contiene el api en el repositorio).

- Creación de la imagen API y validación de su ejecución

`cd proyecto_despliegue_soluciones/api/`

```

ubuntu@ip-172-31-81-62:~/proyecto_despliegue_soluciones$ cd api/
ubuntu@ip-172-31-81-62:~/proyecto_despliegue_soluciones/api$
sudo docker build -t flights-delay-api:latest.
ubuntu@ip-172-31-81-62:~/proyecto_despliegue_soluciones/api$ sudo docker build -t flights-delay-api:latest .
[+] Building 67.1s (13/13) FINISHED
=> [internal] load .dockerignore
=> => transferring context: 28
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 672B
=> [internal] load metadata for docker.io/library/python:3.10

```

`sudo docker images`

```

ubuntu@ip-172-31-81-62:~/proyecto_despliegue_soluciones/api$ sudo docker images
REPOSITORY          TAG         IMAGE ID      CREATED       SIZE
flights-delay-api    latest      30e5bc9412df  45 seconds ago  1.58GB
hello-world          latest      9c7a54a9a43c  6 months ago  13.3kB
ubuntu@ip-172-31-81-62:~/proyecto_despliegue_soluciones/api$

```

- Puede realizar pruebas de ingreso de la imagen y su correcto funcionamiento con IP Publica y puerto 8001

`sudo docker run -p 8001:801 -it -e PORT=8001 flights-delay-api`

- Creación de la imagen del TABLERO y validación de su ejecución

`cd proyecto_despliegue_soluciones/client/`

`sudo docker build -t flights-delay-dashboard:latest.`

`sudo docker images`

- Puede realizar pruebas de ingreso de la imagen y su correcto funcionamiento con IP Publica y puerto 8050 (en esta ejecución la predicción no funcionará pues no se ha configurado el consumo del API desde el tablero, lo haremos más adelante).

`sudo docker run -p 8050:8050 -it -e PORT=8050 flights-delay-dashboard`

### 3. Instalar AWS CLI y Configurar Credenciales en la máquina virtual

- Actualice el índice de paquetes

`sudo apt-get update`

- Como los archivos se encuentran en formato es necesario instalar Zip

`sudo apt install zip unzip`

- Copie localmente los instaladores de la interfaz de línea de comandos de AWS (CLI)

`curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"`

```
ubuntu@ip-172-31-81-62:~/proyecto_despliegue_soluciones/api$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 56.5M 100 56.5M 0 0 105M 0 --:--:-- --:--:-- --:--:-- 105M
```

- Descomprima el archivo y ejecute el instalador

`unzip awscliv2.zip`

`sudo ./aws/install`

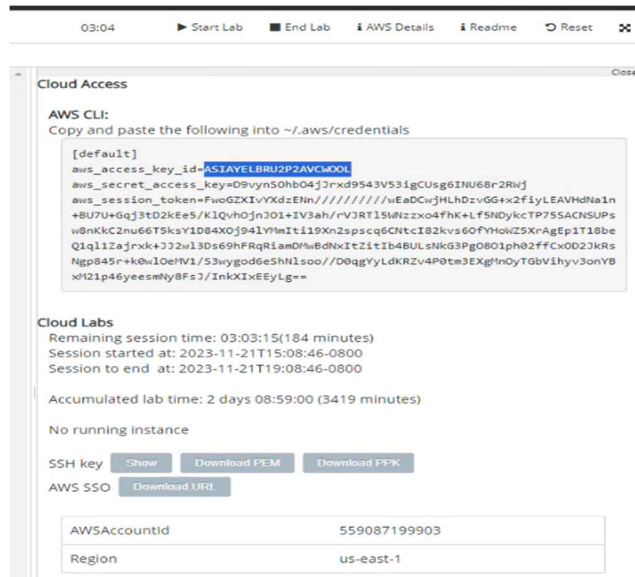
```
ubuntu@ip-172-31-81-62:~/proyecto_despliegue_soluciones/api$ sudo ./aws/install
You can now run: /usr/local/bin/aws --version
```

- Verifique la versión de AWS CLI

`aws --version`

```
ubuntu@ip-172-31-81-62:~/proyecto_despliegue_soluciones/api$ aws --version
aws-cli/2.13.37 Python/3.11.6 Linux/6.2.0-1012-aws exe/x86_64.ubuntu.22 prompt/off
```

- Configure la llave de acceso y la llave secreta de AWS con el comando incluya las llaves asignadas dentro de la consola AWS Details



`aws configure`

```
ubuntu@ip-172-31-81-62:~/proyecto_despliegue_soluciones/api$ aws configure
AWS Access Key ID [None]: ASIAYELBRU2P2AVCW00L
AWS Secret Access Key [None]: D9vynS0hb04jJrxd9543V53igCUsg6INU68r2RWj
Default region name [None]: us-east-1
Default output format [None]:
ubuntu@ip-172-31-81-62:~/proyecto_despliegue_soluciones/api$ _
```

- Configure ahora el token de la sesión con el comando

`aws configure set aws_session_token SESSIONTOKEN`

- Verifique que su cuenta quedó bien configurada solicitando el listado de usuarios de IAM

`aws iam list-users`

```
ubuntu@ip-172-31-81-62:~/proyecto_despliegue_soluciones/api$ aws configure set aws_session_token FuoGZXIvYXdzEln//////////wEaDCwJHLhDzvgG1x2filyLEAVHdNain+BU7U4Gqj3tD2kE5/k1QvOjnj01+Iv3ah/rVJRT15WHzzo4fhk+Lf5NDykTP75SACNSUPsw8NkkC2nu66T5ksY1D84X0j94lYmIti19Xn2spscq6CNtcI82kvs60fYHoWZ5XrAgEp1T18beQ1v4P0tm3EXgfnOyTGBVihyv3onYBxH21p46yeemNy8FsJ/InkXIxEyLg==
ubuntu@ip-172-31-81-62:~/proyecto_despliegue_soluciones/api$ aws iam list-users
{
  "Users": []
}
ubuntu@ip-172-31-81-62:~/proyecto_despliegue_soluciones/api$ _
```

#### 4. Configuración y Despliegue en Amazon ECR

Vaya a la consola de AWS; en la esquina superior izquierda (Servicios) busque el servicio ECR o Elastic Container Registry; cree un repositorio privado; éste es el servicio de registro de imágenes para contenedores; una vez cread copie la URI que será utilizada.

**Recuerde que se debe crear un repositorio para API y otro para TABLERO con los mismos que se indicados:**

The screenshot shows the 'Crear repositorio' (Create repository) page in the AWS ECR console. The 'Configuración general' (General configuration) section is active. Under 'Configuración de visibilidad' (Visibility configuration), the 'Privado' (Private) option is selected. The 'Nombre del repositorio' (Repository name) field contains 'api'. A note below the field states: '3 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.' The 'Inmutabilidad de etiqueta' (Tag immutability) section has 'Desactivado' (Deactivated) selected. A warning message at the bottom says: 'Una vez que se crea un repositorio, la configuración de visibilidad del repositorio no se puede cambiar.'

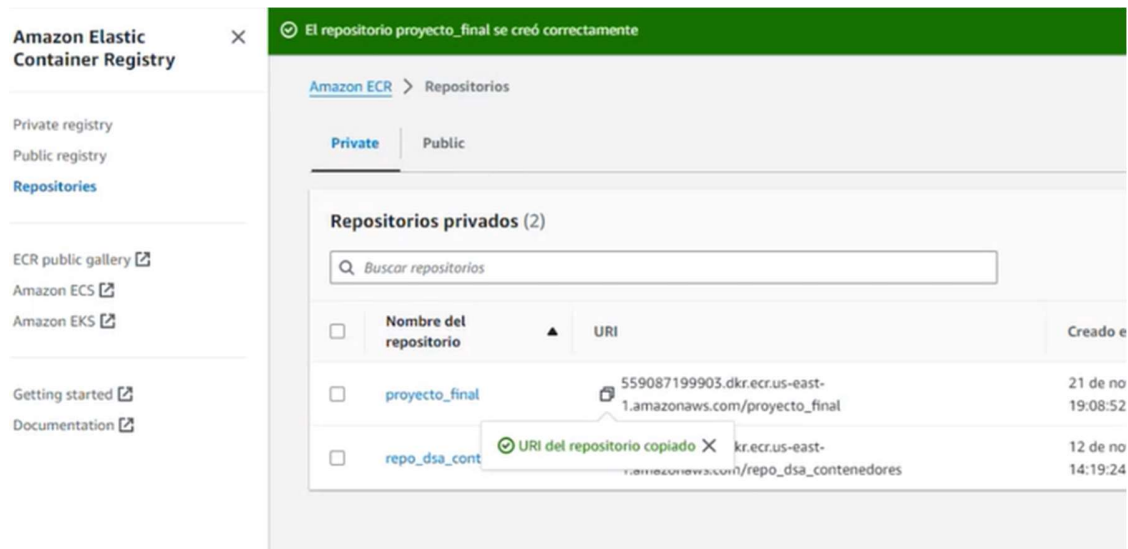
This screenshot is identical to the previous one, but the 'Nombre del repositorio' field contains 'tablero' instead of 'api'. The rest of the configuration remains the same.

The screenshot shows the 'Repositorios' (Repositories) page in the AWS ECR console. A green banner at the top says 'El repositorio tablero se creó correctamente' (The tablero repository was created successfully). The 'Private' tab is selected, showing a list of 4 private repositories. The table columns are: Nombre del repositorio, URI, Creado en, Inmutabilidad de etiqueta, Frecuencia de análisis, Tipo de cifrado, and Caché de extracción.

Nombre del repositorio	URI	Creado en	Inmutabilidad de etiqueta	Frecuencia de análisis	Tipo de cifrado	Caché de extracción
api	559087199903.dkr.ecr.us-east-1.amazonaws.com/api	21 de noviembre de 2023, 19:23:36 (UTC-05)	Desactivado	Manual	AES-256	Inactivo
proyecto_final	559087199903.dkr.ecr.us-east-1.amazonaws.com/proyecto_final	21 de noviembre de 2023, 19:08:52 (UTC-05)	Desactivado	Manual	AES-256	Inactivo
repo_dsa_contenedores	559087199903.dkr.ecr.us-east-1.amazonaws.com/repo_dsa_contenedores	12 de noviembre de 2023, 14:19:24 (UTC-05)	Desactivado	Manual	AES-256	Inactivo
tablero	559087199903.dkr.ecr.us-east-1.amazonaws.com/tablero	21 de noviembre de 2023, 19:23:41 (UTC-05)	Desactivado	Manual	AES-256	Inactivo

## 5. De vuelta en la máquina virtual:

Ahora debemos loguearnos en el registro de AWS ECR para subir las imágenes de las API y el Tablero creadas para lo cual se ejecuta el comando con la URI copiada anteriormente en cada uno de los repositorios creados.



## ❖ API:

- Login ECR del API

`aws ecr get-login-password --region us-east-1 | sudo docker login --username AWS --password-stdin repoURIAPI`

```
ubuntu@ip-172-31-81-62:~$ aws ecr get-login-password --region us-east-1 | sudo docker login --username AWS --password-stdin 559087199903.dkr.ecr.us-east-1.amazonaws.com/proyecto_final
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store
Login Succeeded
```

- Podemos validar que las imagenes previamente generadas del Tablero y el API se encuentran construidas

`sudo docker images`

```
ubuntu@ip-172-31-81-62:~$ sudo docker images
REPOSITORY          TAG         IMAGE ID      CREATED        SIZE
flights-delay-dashboard  latest     4f2152d97e1b  2 minutes ago  1.31GB
flights-delay-api      latest     30e5bc9412df  14 minutes ago  1.58GB
```

- Tag del API, se realiza la etiqueta de la imagen creada con la URI del repositorio en ECS tenga en cuenta la URI copiada para este repositorio

`sudo docker tag flights-delay-api:latest repoURIAPI`

`sudo docker images`

- Carga del API al ECR, se envia la imagen del contenedor al registro de AWS ECR con el comando tenga en cuenta la URI copiada para este repositorio

`sudo docker push repoURIAPI`



```

ubuntu@ip-172-31-81-62:~$ aws ecr get-login-password --region us-east-1 | sudo docker login --username AWS --password-stdin 559087199903.dkr.ecr.us-east-1.amazonaws.com/api
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
ubuntu@ip-172-31-81-62:~$ sudo docker tag flights-delay-api:latest 559087199903.dkr.ecr.us-east-1.amazonaws.com/api
ubuntu@ip-172-31-81-62:~$ sudo docker push 559087199903.dkr.ecr.us-east-1.amazonaws.com/api
Using default tag: latest
The push refers to repository [559087199903.dkr.ecr.us-east-1.amazonaws.com/api]
03a740bef423: Pushed
157be667e9e7: Pushed
a3ea74cad23f: Pushing [=====>] 528MB/551.8MB
a3ea74cad23f: Pushing [=====>] 531.8MB/551.8MB
a3ea74cad23f: Pushing [=====>] 533.9MB/551.8MB
a3ea74cad23f: Pushed [=====>] 562.7MB
0b582dde5b32: Pushed
0dec43bdcdb: Pushed
ab93482a9cdb: Pushed
2b4cf8a5b5de: Pushed
80bd043d4663: Pushed
30f5cd833236: Pushed
7c32e0e08151: Pushed
7cea17427f83: Pushed

latest: digest: sha256:0f82ca2b4ffc73166ca9e0d6af22fbd52c47d50c2c7abdfcf921693540e0a4a size: 3475

```

## ❖ Tablero:

- Login a ECR del TABLERO, con la URI copiada de este repositorio se accede  
[aws ecr get-login-password --region us-east-1 | sudo docker login --username AWS --password-stdin repoURITABLERO](#)
- Tag del tablero, tenga presente la URI del repositorio del tablero creado en los pasos anteriores  
[sudo docker tag flights-delay-dashboard:latest repoURITABLERO](#)
- Carga del tablero al ECR, tenga en cuenta la URI copiada para este repositorio  
[sudo docker push repoURITABLERO](#)

```

ubuntu@ip-172-31-81-62:~$ aws ecr get-login-password --region us-east-1 | sudo docker login --username AWS --password-stdin 559087199903.dkr.ecr.us-east-1.amazonaws.com/tablero
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

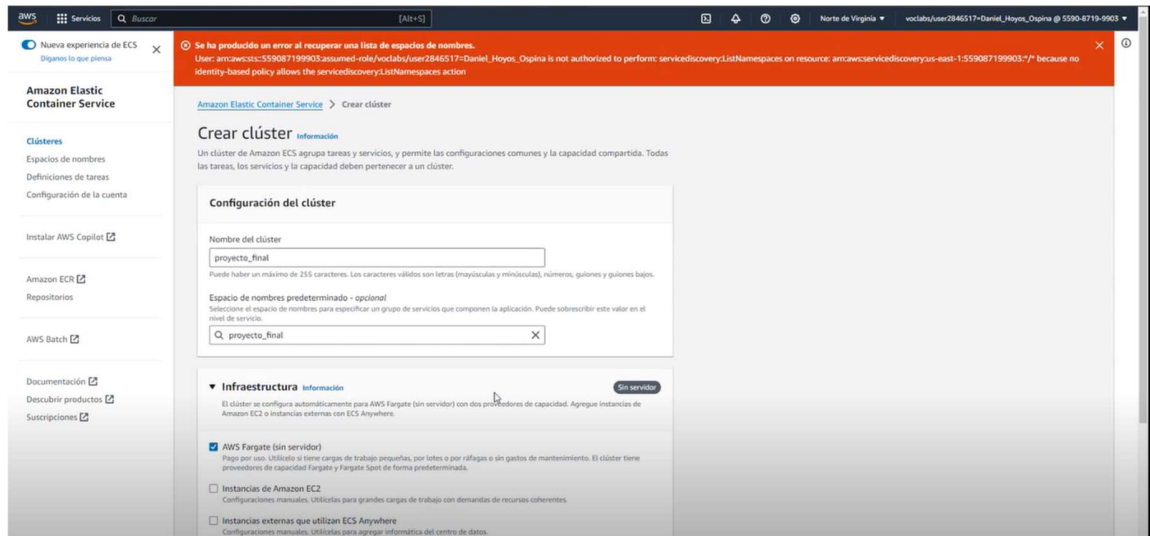
Login Succeeded
ubuntu@ip-172-31-81-62:~$ sudo docker tag flights-delay-dashboard:latest 559087199903.dkr.ecr.us-east-1.amazonaws.com/tablero
ubuntu@ip-172-31-81-62:~$ sudo docker push 559087199903.dkr.ecr.us-east-1.amazonaws.com/tablero
Using default tag: latest
The push refers to repository [559087199903.dkr.ecr.us-east-1.amazonaws.com/tablero]
e20b08167693: Pushed
917376b59fe6: Pushed
c17215898545: Pushing [=====>] 122.9MB/176MB
c17215898545: Pushing [=====>] 125MB/176MB
c17215898545: Pushed
b051f9ec17746: Pushed
4f3517e83d58: Pushed
0b582dde5b32: Pushed
0dec43bdcdb: Pushed
ab93482a9cdb: Pushing [=====>] 43.18MB
2b4cf8a5b5de: Pushed
ab93482a9cdb: Pushed
80bd043d4663: Pushing [=====>] 74.86MB/587.2MB
80bd043d4663: Pushed
30f5cd833236: Pushed
7c32e0e08151: Pushed
7cea17427f83: Pushed

latest: digest: sha256:2738e750409bb8456b2d7e603cd125a23620154ebd5948076773e9cf20b83a size: 3474

```

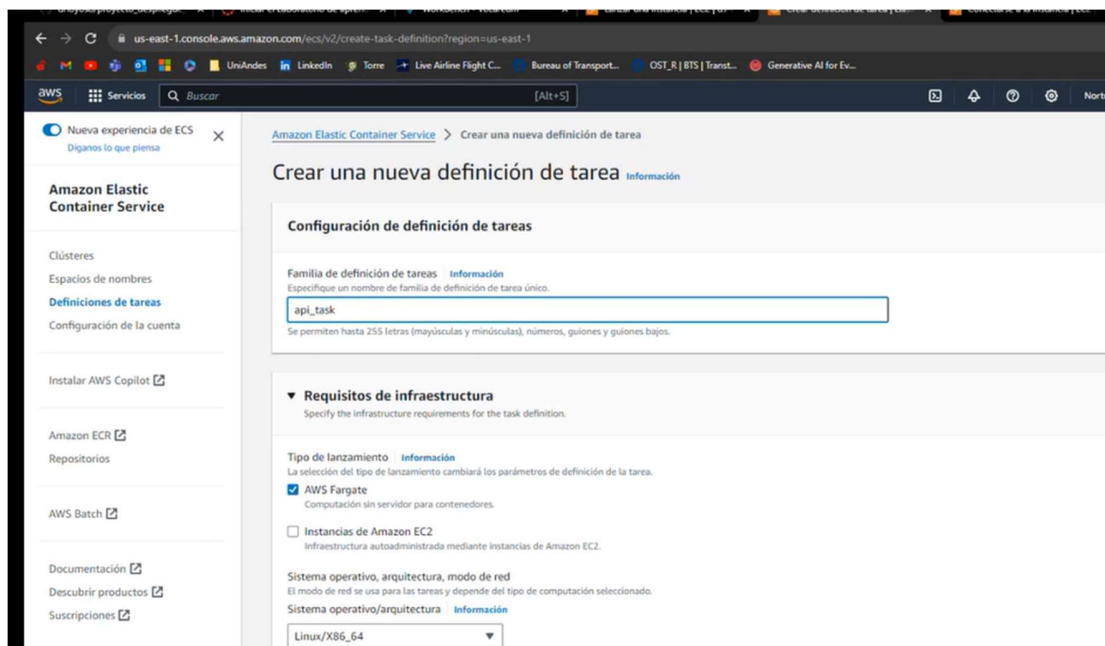
## 6. Crear y Configurar Clúster ECS y Servicios:

- Crear clúster ECS a través de los servicios de la consola AWS



## ❖ Crear definición de tarea y servicio para el API

- Crear una definición de tarea para el API, e n el menú izquierdo vaya a Definiciones de tareas y click en Crear una nueva definición de tarea (Crear definición de tarea), asigne un nombre a la definición de tarea. Como Tipo de lanzamiento seleccione el mismo AWS Fargate.





- En roles de tarea seleccione LabRole (definido por AWSAcademy) y en roles de ejecución de tareas seleccione LabRole (definido por AWSAcademy).

The screenshot shows the AWS Management Console interface for creating a new Amazon ECS task definition. The 'Task role' and 'Execution role' are both set to 'LabRole'. The 'Container' section shows 'Container 1' as the essential container.

- En Contenedor:1 asigne un nombre al contenedor esencial, en URI de la imagen pegue la URI API copiada anteriormente, y marque Sí en Contenedor esencial; en Mapeos de puertos incluya el puerto 8001, las demás características puede dejarlas en sus valores autocompletados y en Límites de recursos fije un límite estricto de memoria de 3 GB.

The screenshot shows the 'Container details' section in the AWS Management Console. It shows the configuration for 'Container 1' with a name 'api', image URI '559087199903.dkr.ecr.us-east-1.amazonaws.com/api:lat', and 'Container essential' set to 'Si'. It also shows port mappings for port 8001 and resource limits for CPU, GPU, and memory.

- En Variable de entorno click en Agregar variable de entorno. En Clave indique PORT y en valor 8001. De esta manera podemos hacer referencia a esta variable en el archivo run.sh (en la carpeta flight\_delays-api).

**Variables de entorno** Información

Agregar individualmente  
Agregue un par clave-valor para especificar una variable de entorno.

Clave	Tipo de valor	Valor	
PORT	Valor	8001	Eliminar

Agregar variable de entorno

Agregar desde archivo  
Agregue variables de entorno de forma masiva al proporcionar un archivo de entorno alojado en Amazon S3.

Agregar archivo de entorno

Puede agregar 10 archivos más de entorno.

- Deje las demás opciones en sus valores por defecto y click en Crear.

► **Monitoreo - opcional**  
Configure los ajustes de rastreo de aplicaciones y recopilación de métricas con la integración de AWS Distro para OpenTelemetry.

► **Etiquetas - opcionales** Información  
Las etiquetas le ayudan a identificar y organizar las definiciones de tareas.

Cancelar **Crear**

- Creamos el servicio del API que utiliza la definición de tarea del API, regrese ahora al Clúster seleccione Servicios y click en Crear servicio.

**Amazon Elastic Container Service** > Clústeres > proyecto\_final > Servicios

proyecto\_final

Información general sobre el clúster

ARN arn:aws:ecs:us-east-1:559087199903:cluster/proyecto_final	Estado Activo	Supervisión de CloudWatch Valor predeterminado	Instancias de contenedor -
Servicios Vacío	Activo	Tareas Pendiente	Ejecutando

Servicios (0) Información

Filtrar tipo de lanzamiento: Cualquier tipo de lanzamiento

Filtrar tipo de servicio: Cualquier tipo de servicio

Nombre del servicio	Estado	ARN	Tipo de se...	Implementaciones y tareas	Última impl...
No hay servicios No hay servicios que mostrar.					

**Crear**

- En la Configuración de implementación seleccione Servicio y en Familia seleccione la definición de tarea recién creada. La revisión debe actualizarse a la más reciente.

Nueva experiencia de ECS  
Díganos lo que piensa

Amazon Elastic Container Service

Clústeres  
Espacios de nombres  
Definiciones de tareas  
Configuración de la cuenta

Instalar AWS Copilot

Amazon ECR  
Repositorios

AWS Batch

Documentación  
Descubrir productos  
Suscripciones

Versión de la plataforma **Información**  
Especifique la versión de la plataforma en la que desea ejecutar el servicio.  
LATEST

**Configuración de implementación**

**Tipo de aplicación** **Información**  
Especifique el tipo de aplicación que desea ejecutar.

☒ **Servicio**  
Lance un grupo de tareas que gestionen un trabajo informático de ejecución prolongada que se pueda detener y reiniciar. Por ejemplo, una aplicación web.

☐ **Tarea**  
Lance una tarea independiente que se ejecute y finalice. Por ejemplo, un trabajo por lotes.

**Definición de tarea**  
Seleccione una definición de tarea existente. Para crear una nueva definición de tarea, vaya a [Definiciones de tareas](#).

☐ **Especificar la revisión manualmente**  
Ingrese manualmente la revisión en lugar de elegir entre las 100 revisiones más recientes para la familia de definición de tareas seleccionada.

Familia: api\_task

Revisión: 1 (MÁS RECIENTE)

**Nombre del servicio**  
Asigne un nombre único a este servicio.

**Tipo de servicio** **Información**  
Especifique el tipo de servicio que seguirá el programador de servicio.

☒ **Replica**  
Coloque y mantenga un número deseado de tareas en su clúster.

☐ **Daemon**  
Coloque y mantenga una copia de la tarea en cada instancia del contenedor.

- Asigne un nombre al servicio y deje las demás opciones en su valor por defecto y click en Crear.

Nueva experiencia de ECS  
Díganos lo que piensa

Amazon Elastic Container Service

Clústeres  
Espacios de nombres  
Definiciones de tareas  
Configuración de la cuenta

Instalar AWS Copilot

Amazon ECR

Versión de la plataforma **Información**  
Especifique la versión de la plataforma en la que desea ejecutar el servicio.  
LATEST

**Configuración de implementación**

**Tipo de aplicación** **Información**  
Especifique el tipo de aplicación que desea ejecutar.

☒ **Servicio**  
Lance un grupo de tareas que gestionen un trabajo informático de ejecución prolongada que se pueda detener y reiniciar. Por ejemplo, una aplicación web.

☐ **Tarea**  
Lance una tarea independiente que se ejecute y finalice. Por ejemplo, un trabajo por lotes.

**Definición de tarea**  
Seleccione una definición de tarea existente. Para crear una nueva definición de tarea, vaya a [Definiciones de tareas](#).

☐ **Especificar la revisión manualmente**  
Ingrese manualmente la revisión en lugar de elegir entre las 100 revisiones más recientes para la familia de definición de tareas seleccionada.

Familia: api\_task

Revisión: 1 (MÁS RECIENTE)

**Nombre del servicio**  
Asigne un nombre único a este servicio.  
api\_service

**Tipo de servicio** **Información**  
Especifique el tipo de servicio que seguirá el programador de servicio.

☒ **Replica**  
Coloque y mantenga un número deseado de tareas en su clúster.

☐ **Daemon**  
Coloque y mantenga una copia de la tarea en cada instancia del contenedor.

- Al regresar a la ventana del Clúster, en el panel de Servicios debe observar el Servicio en ejecución.
- El lanzamiento toma unos minutos. Cuando termine de click en el nombre del servicio.
- Ver en el security group asociado al servicio si el puerto 8001 está abierto y sino, habilitarlo, para lo cual seleccione la pestaña Redes y click en el Grupo de seguridad. En la página del grupo de seguridad, seleccione Editar reglas de entrada y habilite el tráfico por el puerto 8001 desde toda dirección IPv4.

Editar reglas de entrada

Las reglas de entrada controlan el tráfico entrante que puede llegar a la instancia.

ID de la regla del grupo de seguridad	Tipo	Protocolo	Intervalo de puertos	Origen	Descripción: opcional	Acción
sg-08042727a61c4258	Todo el tráfico	Todo	Todo	Personaliza...		Eliminar
sg-07b564d99718b3b26	TCP personalizado	TCP	8001	Personaliza...		Eliminar
-	TCP personalizado	TCP	8050	Anywhere...		Eliminar

Agregar regla

Rules with source of 0.0.0.0 or /0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancelar Previsualizar los cambios Guardar reglas

- Verificar y obtener la IP pública para el API

Amazon Elastic Container Service > Clústeres > proyecto\_final > Servicios > api\_service > Tareas > e33a7d46ff134a5a89039ee4916998fe > Configuración

e33a7d46ff134a5a89039ee4916998fe

Configuración Registros Redes Etiquetas

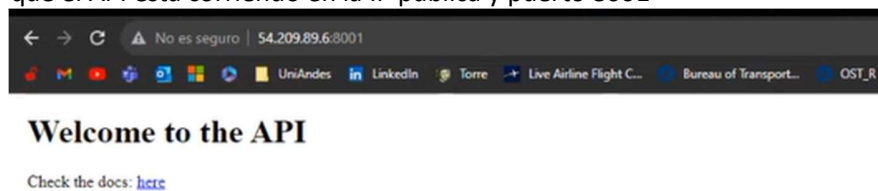
Información general sobre la tarea

ARN arn:aws:ecs:us-east-1:559087199903:task/proyecto_final/e33a7d46ff134a5a89039ee4916998fe	Último estado Ejecutando	Estado deseado Ejecutando	Iniciado/creado a las 2023-11-22T00:37:04.180Z 2023-11-22T00:36:22.991Z
--	-----------------------------	------------------------------	---

Configuración

Sistema operativo/arquitectura Linux/x86_64	Proveedor de capacidad FARGATE	ID de ENI eni-04dc369a898773d0	IP pública copiado/s 54.209.89.6   dirección abierta
CPU   Memoria 1 vCPU   3 GB	Tipo de lanzamiento FARGATE	Modo de red awsvpc	IP privada 172.31.46.187
Versión de la plataforma 1.4.0	Definición de tarea: revisión api_task:1	ID de subred subnet-0663b49d6d6cf9375	Dirección MAC 0e:35:d7:c2:1a:dd
	Grupo de tareas service:api_service		

- Validar que el API está corriendo en la IP pública y puerto 8001



## ❖ Crear definición de tarea y servicio para el Tablero, configurando puerto 8050 y variables de entorno

- Con los pasos de la tarea api realizar la creación de la tarea teniendo en cuenta la IP de la imagen del tablero creada en pasos anteriores y configurar el puerto 8050. Adicionalmente para esta tarea debemos poner una variable de entorno adicional llamada "API\_URL" que contiene la IP pública de la tarea desplegada en contenedor previamente con el API.

Mapas de puertos - [Información](#)

Agregue mapeos de puertos para permitir que el contenedor obtenga acceso a los puertos del host para enviar o recibir tráfico. Cualquier cambio en la configuración de mapeos de puertos afecta a la configuración de conexión del servicio asociada.

Puerto del contenedor	Protocolo	Nombre del puerto	Protocolo de la aplicación	
8050	TCP	dashboard-8050-tcp	HTTP	<a href="#">Eliminar</a>

[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
0	1	3	1
en vCPU		en GB	en GB

**Variables de entorno - opcional**

**Variables de entorno** [Información](#)

**Agregar individualmente**

Agregue un par clave-valor para especificar una variable de entorno.

Clave	Tipo de valor	Valor	
API_URL	Valor	54.209.89.6	<a href="#">Eliminar</a>
PORT	Valor	8050	<a href="#">Eliminar</a>

[Agregar variable de entorno](#)

**Agregar desde archivo**

Agregue variables de entorno de forma masiva al proporcionar un archivo de entorno alojado en Amazon S3.

[Agregar archivo de entorno](#)

Puede agregar 10 archivos más de entorno.

- Con los pasos indicados anteriormente se crea el servicio del tablero teniendo en cuenta la tarea creada para este y seleccionándola en la familia.

Versión de la plataforma [Información](#)

Especifique la versión de la plataforma en la que desea ejecutar el servicio.

LATEST

**Configuración de implementación**

**Tipo de aplicación** [Información](#)

Especifique el tipo de aplicación que desea ejecutar.

☒ Servicio

Lance un grupo de tareas que gestionen un trabajo informático de ejecución prolongada que se pueda detener y reiniciar. Por ejemplo, una aplicación web.

☐ Tarea

Lance una tarea independiente que se ejecute y finalice. Por ejemplo, un trabajo por lotes.

**Definición de tarea**

Seleccione una definición de tarea existente. Para crear una nueva definición de tarea, vaya a [Definiciones de tareas](#).

☐ Especificar la revisión manualmente

Ingrese manualmente la revisión en lugar de elegir entre las 100 revisiones más recientes para la familia de definición de tareas seleccionada.

Familia: dashboard\_task

Revisión: 1 (MÁS RECIENTE)

Nombre del servicio

Asigne un nombre único a este servicio.

dashboard\_service

**Tipo de servicio** [Información](#)

Especifique el tipo de servicio que seguirá el programador de servicio.

☒ Réplica

Coloque y mantenga un número deseado de tareas en su clúster.

☐ Daemon

Coloque y mantenga una copia de la tarea en cada instancia del contenedor.

**Tareas deseadas**

Especifique el número de tareas que se van a lanzar.

- Verificar y abrir el puerto 8050 en el security group asociado al servicio del Tablero de acuerdo con los pasos indicados en esta actividad para el API.

Amazon Elastic Container Service > Clústeres > proyecto\_final > Servicios > dashboard\_service > Redes

dashboard\_service Información

Estado y métricas | Tareas | Registros | Implementaciones | Eventos | Configuración | **Redes** | Etiquetas

**Configuración de red**

Red ⌵	Grupos de seguridad sg-0a992c385e30882a2 <a href="#">🔗</a> Asignar automáticamente una IP pública 🟢 <b>Activado</b> Periodo de gracia de comprobación de estado -	Rol de servicio AWSServiceRoleForECS <a href="#">🔗</a>	Equilibradores de carga - Nombres DNS - Grupos de destino -
----------	--	---	--

VPX > Grupos de seguridad > sg-0a992c385e30882a2 - default > Editar reglas de entrada

Editar reglas de entrada Información

Las reglas de entrada controlan el tráfico entrante que puede llegar a la instancia.

**Reglas de entrada** Información

ID de la regla del grupo de seguridad	Tipo <small>Información</small>	Protocolo <small>Información</small>	Intervalo de puertos <small>Información</small>	Origen <small>Información</small>	Descripción: opcional <small>Información</small>	
sg-08042f27ea61c4258	Todo el tráfico	Todo	Todo	Personaliza... sg-0a992c385e30882a2 <a href="#">✕</a>		Eliminar
sg-07b56de0971bb3b26	TCP personalizado	TCP	8001	Personaliza... 0.0.0.0/0 <a href="#">✕</a>		Eliminar
-	TCP personalizado	TCP	8050	Anywhere... Personalizada Anywhere-IPv4 ✓ Anywhere-IPv6 Mi IP	0.0.0.0/0 <a href="#">✕</a> 0.0.0.0/0 <a href="#">✕</a>	Eliminar

⚠️ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancelar Previsualizar los cambios **Guardar reglas**

- Validar que el Tablero está corriendo en la IP pública del servicio en el puerto 8050

Nueva experiencia de ECS [✕](#)

Amazon Elastic Container Service

Clústeres  
Espacios de nombres  
Definiciones de tareas  
Configuración de la cuenta  
Instalar AWS Copilot [🔗](#)  
Amazon ECR [🔗](#)  
Repositorios  
AWS Batch [🔗](#)  
Documentación [🔗](#)  
Descubrir productos [🔗](#)  
Suscripciones [🔗](#)

816958d1e060488e95e73c805584fccb

Configuración | Registros | Redes | Etiquetas

**Información general sobre la tarea**

ARN arn:aws:ecs:us-east-1:559087199903:task/proyecto_final/816958d1e060488e95e73c805584fccb	Último estado 🟢 <b>Ejecutando</b>	Estado deseado 🟢 <b>Ejecutando</b>	Iniciado/creado a las 2023-11-22T01:28:56.716Z 2023-11-22T01:28:02.167Z
--	--------------------------------------	---------------------------------------	---

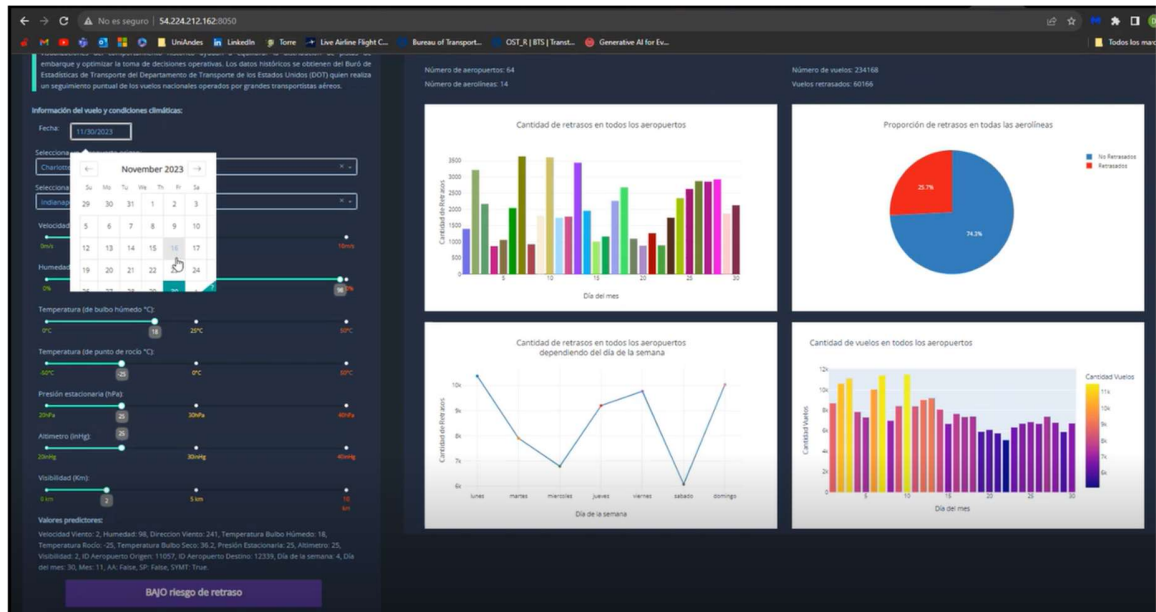
**Configuración**

Sistema operativo/arquitectura Linux/X86_64	Proveedor de capacidad FARGATE	ID de ENI eni-0fe8e9c750fc648c <a href="#">🔗</a>	IP pública 54.224.212.162   dirección abierta <a href="#">🔗</a>
CPU   Memoria 1 vCPU   3 GB	Tipo de lanzamiento FARGATE	Modo de red awsvpc	IP privada 172.31.42.52
Versión de la plataforma 1.4.0	Definición de tareas: revisión dashboard_task-2	ID de subred subnet-0663b49d6d0cf9375 <a href="#">🔗</a>	Dirección MAC 0e:72:90:0b:8a:59
Grupo de tareas service:dashboard_service_3			

**Contenedores (1)**

< 1 > ⌕





Con estos pasos, has desplegado la API y el Tablero en AWS utilizando servicios como ECS y ECR, garantizando escalabilidad y eficiencia en la gestión de retrasos en vuelos comerciales.