

Proyecto

AquaSenseCloud

Infraestructura para la
Computación de Altas Prestaciones

Curso: 2024/2025

Nasim El Arifi Ahmed

Jorge Ballesta Cerezo

Jose Francisco Rabasco Burillo

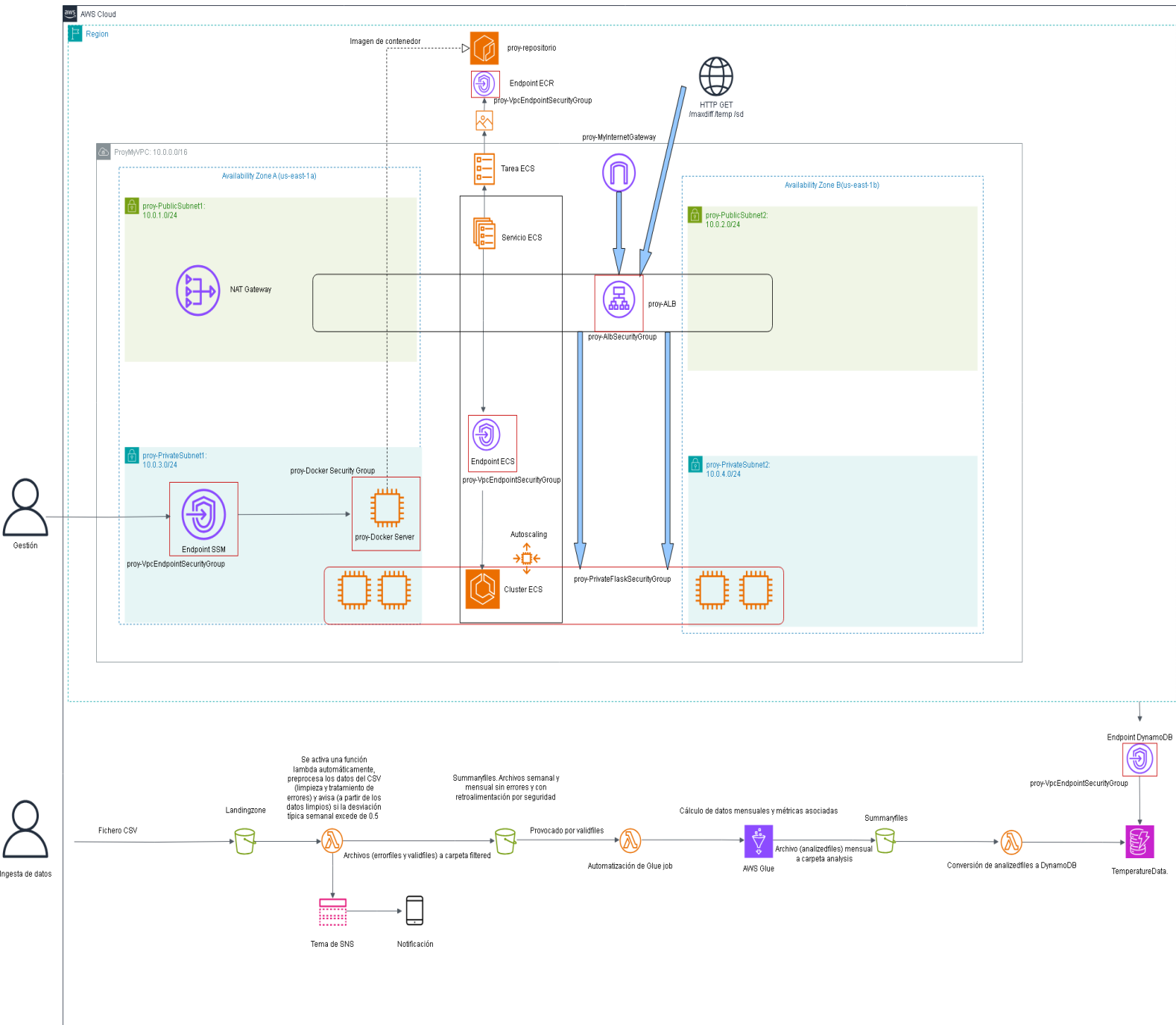
Infraestructura para la Computación de Altas Prestaciones

ÍNDICE

<i>Infraestructura general</i>	3
<i>Plan de trabajo del grupo</i>	4
<i>Tarea 1</i>	5
<i>Tarea 2</i>	7
<i>Tarea 3</i>	9
<i>Tarea 4</i>	10
<i>Tarea 5</i>	14
<i>Resumen de recursos y servicios desplegados</i>	26

Infraestructura para la Computación de Altas Prestaciones

Infraestructura general



Infraestructura para la Computación de Altas Prestaciones

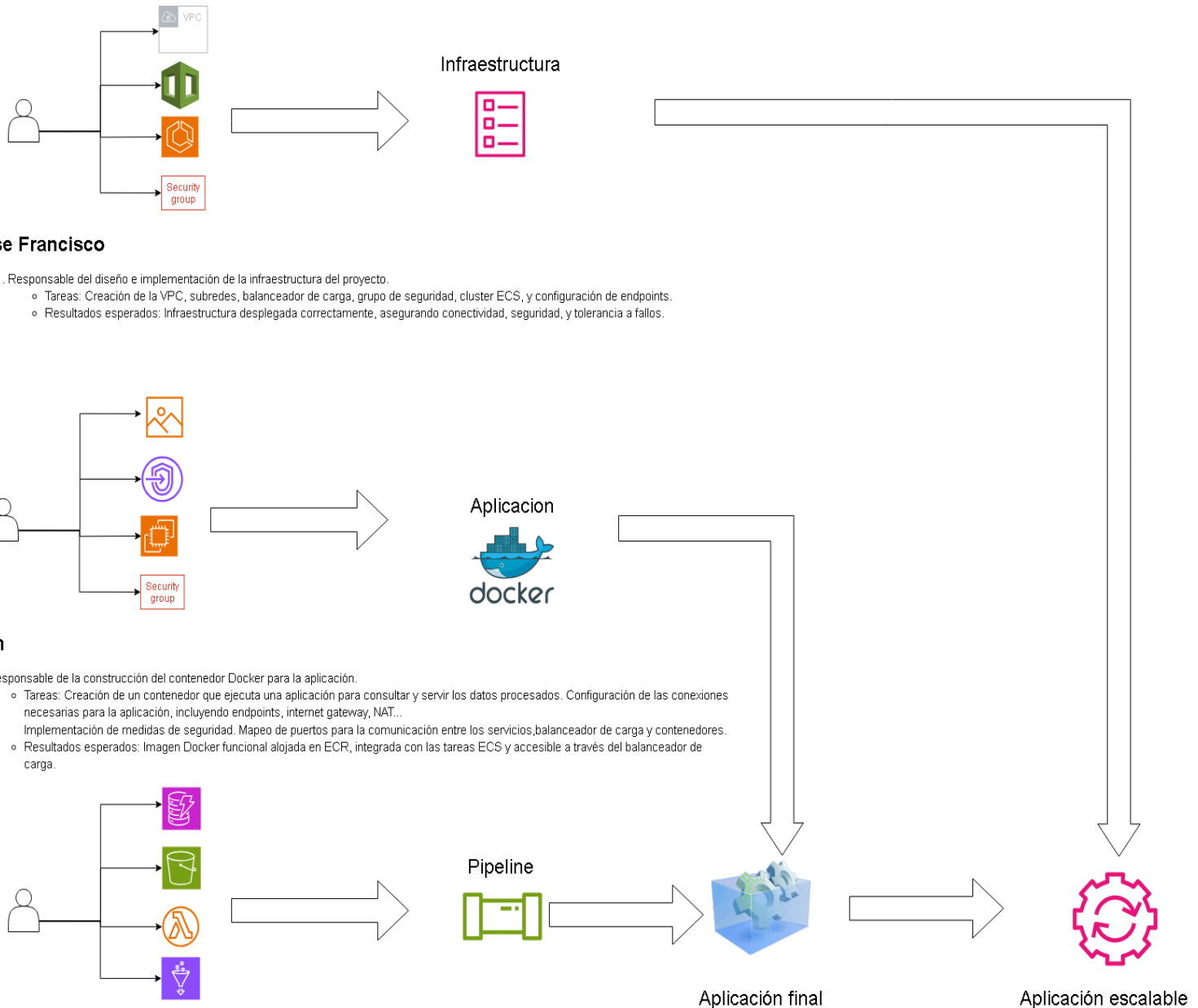
Plan de trabajo del grupo

Plan de Trabajo del Grupo

El equipo se organizó para abordar las diferentes fases del proyecto con una división clara de responsabilidades:

Tarea 1

Explicación y justificación del pipeline.



Infraestructura para la Computación de Altas Prestaciones

Amazon S3

¿Por qué lo elegimos?

- **Almacenamiento económico:** S3 ofrece un modelo de precios basado en el almacenamiento utilizado, lo que permite optimizar costos para grandes volúmenes de datos.
 - **Escalabilidad:** Permite gestionar cantidades masivas de datos, creciendo según sea necesario, sin la necesidad de gestionar infraestructura subyacente.
 - **Durabilidad:** Ofrece una durabilidad del 99.99% y replicación automática entre múltiples zonas de disponibilidad.
 - **Compatibilidad:** Funciona perfectamente con servicios como AWS Glue y Lambda, permitiendo flujos de trabajo integrados.
-

AWS Glue

Funcionamiento:

AWS Glue es un servicio ETL (Extraer, Transformar y Cargar) que simplifica la preparación y transformación de datos para análisis.

¿Por qué lo elegimos?

- **Automatización:** Sus crawlers detectan automáticamente la estructura de los datos, evitando configuraciones manuales tediosas.
 - **Transformación eficiente:** Permite ejecutar trabajos ETL escalables sobre grandes volúmenes de datos utilizando Spark.
 - **Integración:** Se conecta directamente con S3 y DynamoDB, lo que facilita el flujo de datos en el pipeline.
 - **Flexibilidad:** Soporta una amplia variedad de formatos y estructuras de datos, permitiendo extender el pipeline con nuevos esquemas.
-

Amazon DynamoDB

Funcionamiento:

Amazon DynamoDB es una base de datos NoSQL.

¿Por qué lo elegimos?

- **Disponibilidad en tiempo real:** DynamoDB permite acceso de baja latencia a los datos, ideal para consultas rápidas y escalables desde aplicaciones o APIs.
-

Infraestructura para la Computación de Altas Prestaciones

- **Extensibilidad:** Escala automáticamente según el volumen de datos o la cantidad de solicitudes, sin interrupciones del servicio.
 - **Alta disponibilidad:** Opera en múltiples zonas de disponibilidad de AWS, garantizando un funcionamiento continuo incluso ante fallos.
 - **Optimización de costos:** Ofrece modelos de pago basados en la capacidad provisionada o el uso real, ajustándose a diferentes necesidades.
-

AWS Lambda

Funcionamiento:

AWS Lambda permite ejecutar funciones de código sin aprovisionar ni gestionar servidores

¿Por qué lo elegimos?

- **Eventos automáticos:** Lambda se activa cada vez que un archivo nuevo es cargado en S3, lo que permite procesar los datos de manera inmediata.
 - **Eficiencia:** Solo incurre en costos cuando la función se ejecuta, eliminando costos de servidores permanentes.
 - **Facilidad de integración:** Funciona perfectamente con S3, Glue y DynamoDB, conectando las diferentes piezas del pipeline.
-

Amazon SNS

Funcionamiento:

Amazon Simple Notification Service (SNS) permite enviar mensajes a múltiples destinos, como correos electrónicos, mensajes de texto o aplicaciones, en tiempo real.

¿Por qué lo elegimos?

- **Alertas automáticas:** SNS envía notificaciones cuando se detectan desviaciones o errores en los datos, manteniendo a los usuarios informados.
- **Tiempo real:** Las alertas son inmediatas, lo que facilita responder rápidamente a cualquier incidente.
- **Escalabilidad:** Admite miles de suscriptores y mensajes por segundo, ideal para pipelines que puedan escalar.

Justificación global pipeline

El uso de **Amazon S3** como almacenamiento principal permite gestionar grandes volúmenes de datos en un entorno altamente escalable y económico, ideal para almacenar los archivos CSV

Infraestructura para la Computación de Altas Prestaciones

originales y procesados sin preocuparse por límites de capacidad o costos excesivos. La integración de **AWS Glue** permite transformar y procesar estos datos de manera automática y eficiente, generando agregados mensuales clave para el análisis. Este servicio otorga flexibilidad, ya que se adapta a nuevas estructuras de datos o pipelines adicionales con mínimos cambios en la configuración. Por último, la elección de **Amazon DynamoDB** para almacenar los resultados finales asegura una disponibilidad en tiempo real para consultas rápidas a través de una **API RESTful**. DynamoDB es altamente extensible y soporta un acceso de baja latencia, lo que lo hace perfecto para responder a las solicitudes en milisegundos, incluso bajo cargas elevadas. En conjunto, esta infraestructura proporciona un equilibrio perfecto entre almacenamiento económico, capacidad de escalar según la necesidad y rapidez de acceso, lo que la convierte en la mejor solución para este tipo de pipeline de datos.

Tarea 2

Explicación y justificación de la infraestructura

Creación de la Red en Amazon VPC

Se implementó una VPC con redes privadas y públicas distribuidas en zonas de disponibilidad para mejorar la tolerancia a fallos.

CIDR de la VPC: 10.0.0.0/16, suficiente para manejar un gran número de recursos internos.

Subredes públicas (10.0.1.0/24, 10.0.2.0/24) para recursos accesibles desde Internet, como balanceadores de carga.

Subredes privadas (10.0.3.0/24, 10.0.4.0/24) para alojar tareas ECS que requieren seguridad y acceso restringido.

Internet Gateway: Proporciona conectividad externa a las subredes públicas.

NAT Gateway: Necesario para el autoescalado al utilizar instancias EC2, aunque realmente la aplicación en sí podría funcionar únicamente con el uso de endpoints.

Endpoints de VPC: Configurados para SSM, DynamoDB, ECR, y ECS, lo que permite comunicación privada y segura, reduciendo costos al evitar el uso constante del NAT Gateway.

Configuración de Grupos de Seguridad

Los grupos de seguridad controlan el acceso mediante reglas de entrada y salida para permitir solo el tráfico necesario:

Infraestructura para la Computación de Altas Prestaciones

- Grupo de Seguridad del ALB (proyAlbSecurityGroup)

Puerto 80 (HTTP): Permite tráfico desde cualquier origen para manejar solicitudes HTTP entrantes.

Puerto 443 (HTTPS): Permite tráfico seguro para conexiones cifradas.

Grupo de Seguridad para el Servicio ECS (proyPrivateFlaskSecurityGroup)

Puerto 5000 (HTTP): Permite el tráfico desde el ALB hacia las tareas en ECS para la aplicación Flask.

Esta configuración garantiza que solo el balanceador pueda acceder a las tareas. De este modo aseguramos una mayor fiabilidad y seguridad teniendo un entorno controlado.

- Grupo de Seguridad para la Instancia EC2 (proyDockerSecurityGroup)

Puerto 443 (HTTPS): Para comunicación segura mediante el Agente de AWS Systems Manager (SSM). De este modo podremos realizar actualizaciones en la instancia, pudiendo cambiar la aplicación y actualizando así la imagen del contenedor, sin necesidad de conexión vía SSH.

Puerto 5000: Permite tráfico interno dentro de la VPC para la aplicación.

Puerto 80 (HTTP): Necesario para instalar dependencias, Docker...

Estas reglas aseguran que el acceso externo está restringido a puntos controlados y se minimiza la exposición a Internet.

Cluster ECS para Gestión de Contenedores

El clúster ECS (proy-cluster) gestiona tareas y servicios Dockerizados utilizando instancias EC2 con escalado automático.

Una vez hecha la aplicación y alojada en el repositorio ECR, es hora de escalar los recursos mediante un cluster.

Usaremos instancias EC2 que utilizarán una AMI optimizada para ECS para mejorar la compatibilidad con contenedores.

De esta forma conseguimos auto escalado y balanceo de carga

Application Load Balancer (ALB) distribuye el tráfico entre las tareas saludables.

Usaremos la ruta /health para comprobar el estado de las tareas y redirigir solo tráfico a contenedores operativos. Es importante que esto esté implementado en la aplicación como un endpoint, ya que si no, no podremos utilizar el autoescalado al no poder verificar el estado de las tareas.

Automatización

Infraestructura para la Computación de Altas Prestaciones

Toda la infraestructura está definida como código mediante CloudFormation, permitiendo despliegues rápidos, seguros y repetibles. Hemos separado las plantillas en una de red, que contiene toda la infraestructura necesaria, otra de aplicación, que contiene la instancia inicial con la que crearemos la imagen Docker que después se utilizará, y, finalmente, una plantilla que realiza el autoescalado completo en función de la imagen alojada previamente en el repositorio.

Tarea 3

Razonamiento del Pipeline de Despliegue de contenedor/aplicación

Docker en EC2:

El uso de Docker asegura que la aplicación Flask y sus dependencias se ejecuten de manera consistente en cualquier entorno. La imagen Docker es construida, etiquetada y almacenada en Amazon ECR, permitiendo que se pueda acceder y desplegar rápidamente desde cualquier instancia EC2.

El script de instalación automatiza todos los pasos para preparar la instancia EC2 para ejecutar la aplicación Flask, garantizando que cada despliegue sea consistente y confiable.

Integración con Amazon ECR:

Amazon ECR proporciona un repositorio gestionado para almacenar y distribuir imágenes Docker. Al integrar este servicio, el proceso de despliegue se facilita y la gestión de versiones de las imágenes Docker se vuelve más eficiente.

Redundancia y Recuperación ante Fallos:

El uso de subredes privadas y la configuración de volúmenes adicionales garantiza que la infraestructura esté distribuida y sea resiliente a fallos locales.

La creación de volúmenes adicionales y su vinculación con la instancia EC2 permite la persistencia de datos incluso si la instancia se reinicia o detiene.

Escalabilidad en el Despliegue:

Este enfoque automatiza la construcción y despliegue de la imagen Docker en la instancia EC2, lo que permite una expansión **horizontal** de la infraestructura de forma sencilla y controlada.

La arquitectura está diseñada para integrar fácilmente más instancias EC2 en el futuro si fuera necesario, sin intervención manual significativa.

Eficiencia Operativa:

Infraestructura para la Computación de Altas Prestaciones

Gracias al uso de Docker y ECR, el ciclo de vida de desarrollo, pruebas y producción se hace más eficiente, ya que los contenedores proporcionan un entorno consistente que puede ser replicado fácilmente en diferentes máquinas o entornos. Esto optimiza el tiempo de despliegue y facilita las actualizaciones. Así mismo, la conexión mediante endpoints optimizan los recursos al no tener que depender de NAT, aumentando también la seguridad al evitar exposiciones a internet innecesarias.

Tarea 4

Pasos Seguidos para la Creación de la Infraestructura y Automatización del Despliegue del Clúster ECS

Todo el proceso, desde la definición de la red hasta la configuración de balanceadores y escalado automático, han sido automatizado mediante plantillas de AWS CloudFormation para garantizar consistencia, rapidez y seguridad en el despliegue de la infraestructura.

1. Creación del pipeline.

Todo el flujo, desde la carga inicial en S3 hasta el almacenamiento en DynamoDB, está automatizado utilizando eventos de S3 y funciones Lambda, eliminando tareas manuales y errores humanos.

Amazon S3 como punto de entrada

Amazon S3 actúa como el punto de ingesta inicial del pipeline. Los archivos CSV con los datos semanales se cargan en un bucket configurado específicamente para almacenar los datos brutos. Almacenar los datos brutos es importante para poder añadir nuevas implementaciones en el futuro, ya que si no se almacenaran y se procesaran directamente (ETL) habría que cargar de nuevo todos si deseáramos realizar nuevos análisis.

Automatización con AWS Lambda

Una función Lambda se activa automáticamente cuando un archivo nuevo es cargado en el bucket de entrada de S3. Este evento desencadena la ejecución del pipeline, eliminando la necesidad de intervención manual. Lambda también verifica que el archivo cargado cumpla con los requisitos iniciales, como el formato esperado o la integridad básica. En nuestro caso realiza un preprocesamiento básico de limpieza y normalización de datos.

Amazon S3 como punto de salida filtrada

Después de que Lambda valide y procese los datos, estos se dividen en dos categorías:

Datos válidos: Registros limpios y normalizados.

Datos inválidos: Registros con errores, como valores fuera de rango o fechas mal formateadas.

Ambos conjuntos de datos se almacenan en un bucket S3 dedicado, en una carpeta con los datos filtrados, para mantener una trazabilidad clara. Esta separación permite manejar los datos inválidos

Infraestructura para la Computación de Altas Prestaciones

de forma manual o a través de procesos automáticos adicionales, mientras que los datos válidos están listos para el procesamiento avanzado. Este enfoque asegura la integridad del pipeline y facilita la depuración en caso de inconsistencias.

Procesamiento avanzado con AWS Glue

AWS Glue ejecuta un Job que transforma los datos semanales validados en métricas mensuales. El Job realiza las siguientes tareas:

Agrupar los datos por mes y calcular las métricas, como:

Temperatura media mensual.

Temperatura máxima mensual. (útil solo para comprobación de diferencia de temperatura máxima)

Desviación máxima mensual.

Diferencia de temperatura máxima respecto al mes anterior.

Genera un nuevo archivo CSV consolidado con los resultados mensuales en el bucket de salida dentro de la carpeta analysis.

El archivo procesado se guarda en un bucket S3 dedicado, asegurando que los datos estén listos para su análisis o consumo posterior. AWS Glue fue elegido por su capacidad de escalar automáticamente y manejar grandes volúmenes de datos de manera eficiente.

Almacenamiento de los datos en DynamoDB

El archivo procesado con las métricas mensuales se ingesta en una tabla DynamoDB para permitir consultas rápidas y en tiempo real. DynamoDB es ideal para almacenar datos tabulares con claves de acceso claras, como el Año y Mes. Esto asegura que la aplicación puede acceder a ellos con baja latencia mediante un endpoint. Además, DynamoDB escala automáticamente para manejar diferentes niveles de tráfico, lo que lo hace adecuado para datos dinámicos y sistemas en tiempo real. Dicha tabla es la que será usada por la aplicación para realizar las consultas.

2. Creación de la Red con Amazon VPC

La plantilla proy-red.yaml define una red personalizada para soportar los servicios ECS y otros recursos internos:

VPC (proyMyVPC): Se creó una VPC con un CIDR 10.0.0.0/16 habilitada con soporte para nombres de host y resolución DNS.

Subredes Públicas y Privadas:

Dos subredes públicas:

proyPublicSubnet1: 10.0.1.0/24 en us-east-1a.

Infraestructura para la Computación de Altas Prestaciones

proyPublicSubnet2: 10.0.2.0/24 en us-east-1b.

Dos subredes privadas:

proyPrivateSubnet1: 10.0.3.0/24 en us-east-1a.

proyPrivateSubnet2: 10.0.4.0/24 en us-east-1b.

Las subredes privadas están diseñadas para alojar tareas y servicios sin acceso directo a Internet.

Internet Gateway (proyMyInternetGateway): Permite la conectividad de las subredes públicas a Internet.

NAT Gateway (proyNatGateway): Proporciona acceso temporal a Internet para las subredes privadas, con un Elastic IP. Aumentará el coste, pero también la seguridad. De esta forma optamos por una solución de compromiso.

Endpoints de VPC:

Se configuraron endpoints para servicios esenciales como SSM, ECS, y ECR, lo que permite una comunicación privada y segura sin necesidad de un NAT Gateway continuo. De esta forma optimizamos el ahorro de recursos, al tener una conexión directamente interna a través de la red de AWS.

Tabla de Rutas:

Se crearon tablas de rutas separadas para subredes públicas y privadas, asociando las rutas públicas con el Internet Gateway y las privadas con el NAT Gateway y los endpoints de VPC.

Este diseño asegura alta disponibilidad, seguridad y conectividad eficiente para todos los recursos de red.

Al introducir esta plantilla, tendremos desplegados todos estos recursos que componen la infraestructura necesaria para el proyecto.

3. Implementación de la Solución (Contenedor y App)

Creación de la Instancia EC2: Se definió una instancia EC2 con tipo t2.micro dentro de una subred privada, con Docker preinstalado mediante un script en UserData. Esta instancia sirve como servidor para alojar la aplicación Flask en un contenedor Docker. Aumentamos la seguridad al alojarla en la subred privada y permitimos una futura gestión de desarrollo al habilitar la conexión mediante SSM. La aplicación se crea y se sube directamente al repositorio ECR creado en la pila de red, lo cual es bueno ya que automatiza todo el proceso. Sin embargo, a la hora de realizar cualquier cambio en la instancia (al conectarnos por SSM), es necesario también actualizar esta plantilla para futuros despliegues.

Configuración del Contenedor Docker: Se creó un directorio donde se almacenan los archivos necesarios para la aplicación, incluyendo el archivo Dockerfile, que define cómo se construye la imagen del contenedor. Además, se definieron las dependencias en un archivo requirements.txt. Al arrancar la instancia, se crea la imagen la cual es subida directamente al repositorio ECR.

Infraestructura para la Computación de Altas Prestaciones

Configuración del Código de la Aplicación Flask: Se desarrollaron varios endpoints dentro de la aplicación Flask:

/health: Verificación de la salud del servicio. Es crucial para el autoescalado como hemos explicado anteriormente.

/maxdiff: Obtención de la diferencia máxima de temperatura para un mes y año específicos.

/sd: Obtención de la desviación típica de temperatura.

/temp: Obtención de la temperatura media mensual.

La aplicación interactúa con DynamoDB a través del endpoint definido anteriormente para consultar estos datos y devolver respuestas en formato JSON.

Configuración de Volumen Adicional para la Instancia EC2: Se añadió un volumen EBS de 100 GB para proporcionar almacenamiento adicional a la instancia EC2. Este volumen está vinculado a la instancia y configurado para tomar snapshots durante su eliminación.

Seguridad: Como ya hemos explicado, esta instancia tiene asociada un grupo de seguridad que permite acceso por el puerto 80 (instalar Docker, conexión a través del NAT), 443 (para conexión SSM que permita gestión de instancia) y 5000 (puerto de comunicación para los contenedores).

Una vez ejecutada dicha plantilla, ya tenemos la imagen alojada en el repositorio que hemos creado en la pila de red, por lo que solo nos queda poner en funcionamiento la aplicación y habilitar el auto escalado.

4. Creación del Clúster ECS con Auto Scaling y Balanceo de Carga

La plantilla proy-cluster.yaml automatiza la configuración del clúster ECS para ejecutar aplicaciones contenedorizadas:

Clúster ECS (proy-cluster): Se provisionó un clúster ECS llamado proy-cluster para gestionar tareas y servicios.

Grupo de Auto Scaling (proyEcsAutoScalingGroup):

Inicializa con 4 instancias EC2 del tipo t2.micro.

Permite escalar entre 4 y 6 instancias según las necesidades de carga.

Utiliza una plantilla de lanzamiento (LaunchTemplate) con Amazon Linux optimizado para ECS, integrando la IAM Instance Profile LabInstanceProfile para permisos necesarios.

Balanceador de Carga (proyAlb):

Infraestructura para la Computación de Altas Prestaciones

Se configuró un Application Load Balancer (ALB) en las subredes públicas para distribuir tráfico. Es muy importante disponer de más de una región de disponibilidad para poder balancear la carga, razón por la cual se tiene las subredes en regiones distintas.

Listener en puerto 80 redirige al grupo de destino (proyEcsLoadBalancerTargetGroup) que escucha en el puerto 5000. Por lo tanto, a la hora de acceder mediante la url, simplemente tendremos que poner la URL del balanceador de carga (protocolo http) y la ruta de la consulta deseada.

Comprobación de Salud:

Implementada en la ruta /health para validar que solo las tareas saludables reciban tráfico.

Definición de Tareas (proyEcsTaskDefinition):

Define recursos para ejecutar contenedores usando imágenes de Amazon ECR.

Configuración de CPU, memoria, y compatibilidad con la red awsvpc.

Escalabilidad Automática:

Se define una política de escalado automático basada en el número de solicitudes por segundo, ajustando las tareas según la carga.

Automatización, Seguridad y Escalabilidad

El enfoque basado en CloudFormation permite automatizar la creación de toda la infraestructura como código, minimizando errores manuales y asegurando implementaciones consistentes. Los recursos se escalan automáticamente para optimizar costos, mientras que las configuraciones de seguridad garantizan un acceso controlado solo a las partes necesarias. Este diseño soporta cargas de trabajo dinámicas con alta disponibilidad y tiempos de respuesta bajos.

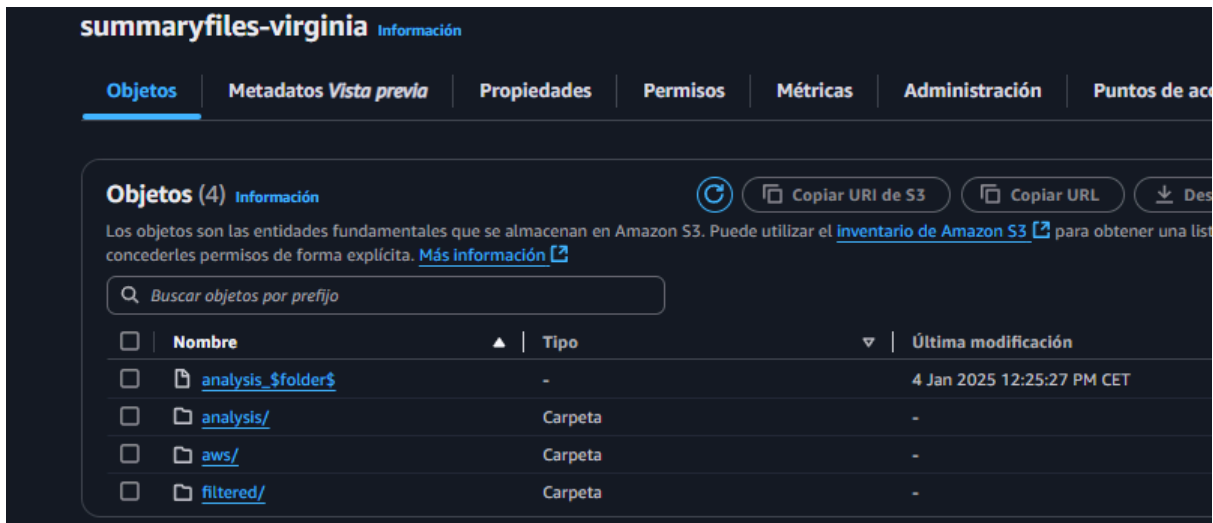
Tarea 5

Ejemplo de despliegue y anexo de replicación

Creación de los buckets S3

Creamos dos buckets tanto para la ingesta como para el almacenamiento de los datos filtrados (/filtered) y los agregados mensuales (/analysis).

Infraestructura para la Computación de Altas Prestaciones



summaryfiles-virginia Información

Objetos | Metadatos *Vista previa* | Propiedades | Permisos | Métricas | Administración | Puntos de acceso

Objetos (4) Información 🔄 Copiar URI de S3 Copiar URL ⬇️ Descargar

Los objetos son las entidades fundamentales que se almacenan en Amazon S3. Puede utilizar el [Inventario de Amazon S3](#) para obtener una lista de objetos y concederles permisos de forma explícita. [Más información](#)

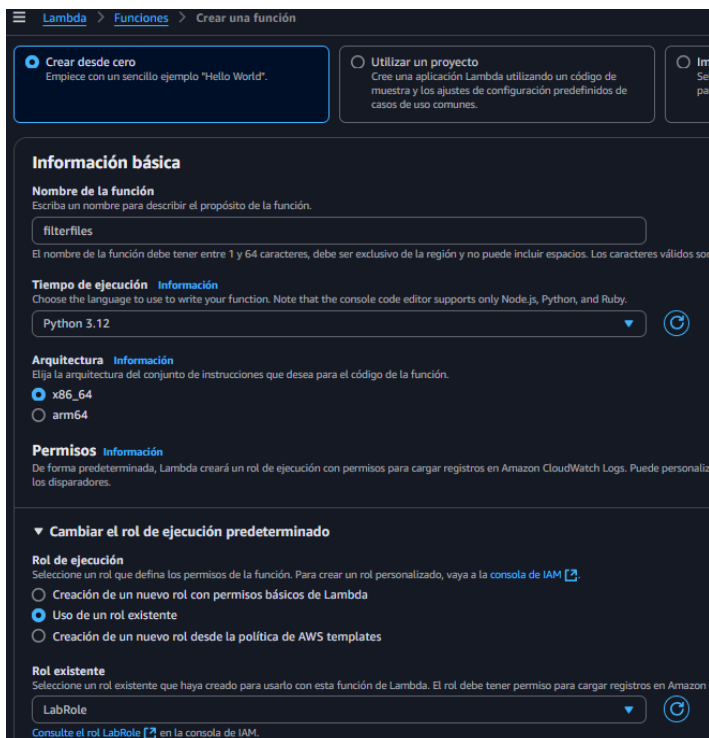
🔍 Buscar objetos por prefijo

<input type="checkbox"/>	Nombre	Tipo	Última modificación
<input type="checkbox"/>	analysis_\$folder\$	-	4 Jan 2025 12:25:27 PM CET
<input type="checkbox"/>	analysis/	Carpeta	-
<input type="checkbox"/>	aws/	Carpeta	-
<input type="checkbox"/>	filtered/	Carpeta	-

Despliegue de las funciones lambda “filterfiles”

Esta función es la encargada de asegurar la integridad de los datos y mantener un formato estándar para todas las entradas. También gestiona errores como valores faltantes o imposibles. Estos datos se almacenan en dos archivos separados que diferencian las entradas que cumplen los estándares y las que no. Para ello, se crean dos archivos .csv que mantienen los registros de forma segura y escalable

Al tratar con una masiva ingesta de datos es necesario modificar la configuración de lambda, aumentando sus recursos y tiempo de espera.



Lambda > Funciones > Crear una función

☒ Crear desde cero
Empezar con un sencillo ejemplo "Hello World".

☐ Utilizar un proyecto
Cree una aplicación Lambda utilizando un código de muestra y los ajustes de configuración predefinidos de casos de uso comunes.

☐ Importar código de un repositorio

Información básica

Nombre de la función
Escriba un nombre para describir el propósito de la función.
filterfiles
El nombre de la función debe tener entre 1 y 64 caracteres, debe ser exclusivo de la región y no puede incluir espacios. Los caracteres válidos son: [a-z0-9-_].

Tiempo de ejecución Información
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Python 3.12

Arquitectura Información
Elija la arquitectura del conjunto de instrucciones que desea para el código de la función.
☒ x86_64
☐ arm64

Permisos Información
De forma predeterminada, Lambda creará un rol de ejecución con permisos para cargar registros en Amazon CloudWatch Logs. Puede personalizar los disparadores.

▼ Cambiar el rol de ejecución predeterminado

Rol de ejecución
Seleccione un rol que defina los permisos de la función. Para crear un rol personalizado, vaya a la [consola de IAM](#).
☐ Creación de un nuevo rol con permisos básicos de Lambda
☒ Uso de un rol existente
☐ Creación de un nuevo rol desde la política de AWS templates

Rol existente
Seleccione un rol existente que haya creado para usarlo con esta función de Lambda. El rol debe tener permiso para cargar registros en Amazon CloudWatch Logs.
LabRole
[Consulte el rol LabRole](#) en la consola de IAM.

Infraestructura para la Computación de Altas Prestaciones

☰ [Lambda](#) > [Funciones](#) > [filterfiles](#) > Editar configuración básica

Editar configuración básica

Configuración básica [Información](#)

Descripción - Opcional

Memoria [Información](#)
La CPU asignada a la función es proporcional a la memoria configurada.

500 MB

Establezca la memoria en un valor entre 128 MB y 10240 MB.

Almacenamiento efímero [Información](#)
Puede configurar hasta 10 GB de almacenamiento efímero (/tmp) para la función. [Ver precios](#)

512 MB

Establezca el almacenamiento efímero (/tmp) entre 512 MB y 10240 MB.

SnapStart [Información](#)
Reduzca el tiempo de inicio haciendo que Lambda almacene en caché una instantánea de la función una vez esta se inicialice. Para evaluar si el caché de instantáneas, consulte las [consideraciones de compatibilidad de SnapStart](#). Para las versiones ejecutables de Python y .NET, consulte los [precios](#).

None

Tiempos de ejecución admitidos: .NET 8 (C#/F#/PowerShell), Java 11, Java 17, Java 21, Python 3.12, Python 3.13.

Tiempo de espera

1 min 0 s

Rol de ejecución
Seleccione un rol que defina los permisos de la función. Para crear un rol personalizado, vaya a la [consola de IAM](#).

☒ Uso de un rol existente

☐ Creación de un nuevo rol desde la política de AWS templates

Rol existente
Seleccione un rol existente que haya creado para usarlo con esta función de Lambda. El rol debe tener permiso para cargar registros en Amazon CloudWatch.

LabRole

[Consulte el rol LabRole](#) en la consola de IAM.



Desencadenador: Actualización o llegada de nuevos archivos al bucket de ingesta (landing_zone)

Destino: Bucket de almacenamiento de datos procesados (summary_files)

En el siguiente paso, desplegaremos los recursos necesarios para una correcta suscripción y notificación.

Pruebas del correcto funcionamiento de filterfiles

Infraestructura para la Computación de Altas Prestaciones

Cargar: estado

Después de salir de esta página, la siguiente información ya no estará disponible.

Resumen

Destino s3://landingzone-virginia	Realizado correctamente 1 archivo, 18.1 KB (100.00%)	Con errores 0 archivos, 0 B (0%)
--------------------------------------	---	-------------------------------------

Archivos y carpetas

Archivos y carpetas (1 total, 18.1 KB)

Buscar por nombre

Nombre	Carpetas	Tipo	Tamaño	Estado
Temperatura.csv	-	text/csv	18.1 KB	Realizado correctamente

Tratamiento de duplicados

140	2020/01/08,12.64578069110484,0.24762567687047482	140	2020/1/8,12.64578069110484,0.24762567687047482
141	2020/01/08,12.64578069110484,0.24762567687047482	141	2020/1/8,12.64578069110484,0.24762567687047482
171	2020/07/23,28.193379487839756,0.27494333121397047	174	2020/7/23,28.193379487839756,0.27494333121397047
		175	2020/7/23,28.193379487839756,0.27494333121397047

filtered/

Objetos

Objetos (2) Información

Los objetos son las entidades fundamentales que se almacenan en Amazon S3. Puede utilizar el [inventario de Amazon S3](#) para obtener una lista de todos los objetos de s3 y concederles permisos de forma explícita. [Más información](#)

Buscar objetos por prefijo

Nombre	Tipo	Última modificación	Tamaño
errorFiles.csv	csv	5 Jan 2025 10:47:14 AM CET	
validFiles.csv	csv	5 Jan 2025 10:47:13 AM CET	

Contenido de validFiles

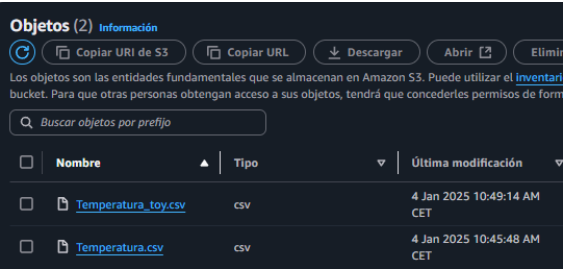


Infraestructura para la Computación de Altas Prestaciones

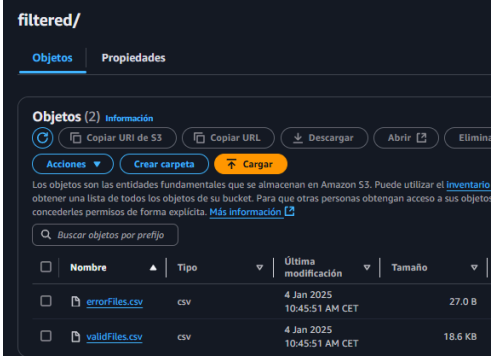
	A	B	C	D
353	2023/11/29,16.144369698571857,0.2107116990915088			
354	2023/12/7,14.622605668678652,0.30889326462476957			
355	2023/12/13,14.899582079008727,0.14463516100329263			
356	2023/12/19,13.217669363232504,0.30770104319446306			
357	2023/12/27,12.54446008930322,0.4683115891884751			
358	2024/1/3,12.549188181281455,0.3736595668933671			
359	2024/1/9,11.428418471222265,0.3528536048794776			
360	2024/1/16,12.29389882095167,0.2260837077596532			
361	2024/1/23,13.256911948083616,0.593878096193998			
362	2024/2/1,13.908418521051017,0.14557727432732773			
363	2024/2/6,14.142304884299088,0.23244666500074718			
364	2024/2/13,14.18089041737649,0.11978314461879211			
365	2024/2/20,15.404827824239252,0.15705175431206714			
366	2024/3/7,14.424544046425073,0.15699769899074903			

Introducimos ahora nuevos datos.

Contenido Temperatura_toy



	A	B	C
1	Fecha,Medida,Desviaciones		
2	2024/4/24,19.6412031576098,0.278		
3	2024/4/29,18.996086724745407,0.74		
4	2024/5/10,21.64996320177442,0.29		
5	2024/5/15,22.46870113165072,0.38		
6	2024/5/22,23.309800739260282,0.57		
7	2024/5/29,24.439307464839267,0.44		
8	2024/6/5,24.994100237465585,0.20		
9	2024/6/10,25.106752427481688,0.32		
10	2024/6/18,26.398364126201148,0.22		
11	2024/6/25,26.74750931996531,0.54		
12	2024/7/3,27.523151354544765,0.20		
13	2024/7/9,28.01059929538777,0.36		
14	2024/7/18,29.458599356758782,0.19		



ValidFiles actualizado respecto al primer archivo.

354	2024/01/03,12.549188181281455,0.3736595668933671
355	2024/01/09,11.428418471222265,0.3528536048794776
356	2024/01/16,12.29389882095167,0.2260837077596532
357	2024/01/23,13.256911948083616,0.593878096193998
358	2024/02/01,13.908418521051017,0.14557727432732773
359	2024/02/06,14.142304884299088,0.23244666500074718
360	2024/02/13,14.18089041737649,0.11978314461879211

Infraestructura para la Computación de Altas Prestaciones

Implementación del SNS

Creamos un tema dentro de SNS y definimos correctamente la suscripción mediante correo electrónico.

The image shows two parts of the AWS SNS console. On the left, the 'Temas (2)' (Topics) list shows two topics: 'RedshiftSNS' and 'SD_LIMIT', with 'SD_LIMIT' circled in red. On the right, the 'Crear una suscripción' (Create a subscription) form is shown. It includes fields for 'ARN del tema' (Topic ARN) with the value 'arn:aws:sns:us-east-1:829511480989:SD_LIMIT', 'Protocolo' (Protocol) set to 'Correo electrónico' (Email), and 'Punto de enlace' (Endpoint) set to 'ballesta.rabasco.elarifi@gmail.com'.

Pruebas del correcto funcionamiento del SNS

<input type="checkbox"/>	★ AWS Notifications 3	Alerta de Desviación Típica - Alerta: Desviación típica de 0.7...	<input type="checkbox"/>
<input type="checkbox"/>	★ AWS Notifications 39	Alerta de Desviación Típica - Alerta: Desviación típica de 0.74 detectad...	<input type="checkbox"/>
<input type="checkbox"/>	★ AWS Notifications 100	Alerta de Desviación Típica - Alerta: Desviación típica de 0.52743977...	<input type="checkbox"/>

Implementación del Glue Job + automatización lambda

Definimos un Glue Job que transforma los datos semanales validados en métricas mensuales.

The image shows two parts of the AWS Glue console. On the left, the 'Basic properties' section shows the job name 'aggregated_values', description, IAM role, and type 'Spark'. On the right, the 'Script' section shows the engine 'Spark' and options 'Start fresh' and 'Upload script'. A 'Choose file' button is visible, with a note 'Limited to Python (*.py, *.py3) files only'.

Una vez lo hemos definido correctamente, está listo para su ejecución. Sin embargo, es necesario implementar una función lambda que automatice el proceso, de manera que se ejecute el script cuando el archivo validfiles.csv es modificado.

Infraestructura para la Computación de Altas Prestaciones

Agregar desencadenador

Configuración del desencadenador Información

S3
aws asynchronous storage

Bucket
Seleccione o escriba el ARN de un bucket de S3 que actúa como origen de eventos. El bucket debe estar en la misma región que la función.
Q s3/summaryfiles-virginia X C
Región del bucket: us-east-1

Tipos de eventos
Seleccione los eventos que desea que activen la función de Lambda. Si lo desea, también puede configurar un prefijo o un sufijo para un evento. Sin embargo, en cada bucket, no puede haber eventos individuales con configuraciones múltiples que tengan prefijos o sufijos superpuestos que puedan coincidir con la misma clave de objeto.
Todos los eventos de creación de objetos X

Prefijo - Opcional
Introduzca un único prefijo opcional para limitar las notificaciones a los objetos cuyas claves comiencen por los caracteres coincidentes. Todos los caracteres especiales deben estar codificados en URL.
filtered/

Sufijo - Opcional
Introduzca un único sufijo opcional para limitar las notificaciones a los objetos cuyas claves terminen por los caracteres coincidentes. Todos los caracteres especiales deben estar codificados en URL.
validFiles.csv

Invocación recurrente
Si la función escribe objetos en un bucket de S3, asegúrese de utilizar diferentes buckets de S3 para las entradas y las salidas. Escribir en el mismo bucket aumenta el riesgo de crear una invocación recurrente, lo que puede resultar en un aumento del uso de Lambda y de los costos. Más información
☒ Acepto que no se recomienda utilizar el mismo bucket de S3 para las entradas y las salidas, y que esta configuración puede provocar invocaciones recurrentes, así como también un aumento del uso de Lambda y de los costos.

Lambda añadirá los permisos necesarios para AWS S3 para invocar la función Lambda desde este desencadenador. Obtenga más información sobre el modelo de permisos de Lambda.

Pruebas del correcto funcionamiento del Glue Job

Una vez se detecta el archivo validfiles, se desencadena la función lambda de manera que se corre el job de manera sistemática.

aggregated_values Last modified on 4/1/2025, 12:05:35 **Actions** **Save**

Script **Job details** **Runs** **Data quality** **Schedules** **Version Control** **Upgrade analysis - preview**

Job runs (1/1) Info Last updated (UTC) January 4, 2025 at 11:12:21 **View details** **Stop job run** **Troubleshoot with AI** **Table View** **Card V**

Q Filter job runs by property

Run status	Retries	Start time (Local)	End time (Local)	Duration	Capacity ...	Worker type
Succeeded	0	01/04/2025 12:05:38	01/04/2025 12:07:39	1 m 49 s	10 DPUs	G.1X

Tras la ejecución el archivo analizado de formato csv será almacenado dentro de la carpeta /analysis del bucket summaryfiles.

<input type="checkbox"/>	Nombre	Tipo	Última modificación	Tamaño	Clase de almacenamiento
<input type="checkbox"/>	analysis/	Carpeta	-	-	-
<input type="checkbox"/>	aws/	Carpeta	-	-	-
<input type="checkbox"/>	filtered/	Carpeta	-	-	-

Datos contenidos dentro de analyzedFiles

Infraestructura para la Computación de Altas Prestaciones

Ano,Mes,TempMediaMensual,TempMaxMensual,DiferenciaTempMax
2017,3,17.056983947753906,17.32989501953125,0.0
2017,4,18.872126897176106,19.46531867980957,2.1354236602783203
2017,5,21.888142903645832,23.128435134887695,3.663116455078125
2017,6,26.325317764282225,28.228496551513672,5.100061416625977
2017,7,28.274396419525146,29.515522003173828,1.2870254516601562
2017,8,27.431928634643555,27.431928634643555,-2.0835933685302734
2017,9,25.909508387247723,27.1295223236084,-0.30240631103515625
2017,10,23.622084617614746,24.87564468383789,-2.253877639770508

Despliegue de la base de datos dynamoDB automatizado con lambda.

Para finalizar la construcción de nuestro pipeline de datos, necesitamos convertir el csv analizado con los datos mensuales a una tabla dynamodb. Para ello implementamos una función lambda basada en la actualización o creación de analyzedFiles que contiene los datos mensuales como hemos observado.

Creamos la tabla TemperatureData cuya clave principal será Año y la clave de ordenación mes (los datos clave en las consultas).

Crear tabla

Detalles de la tabla Información

DynamoDB es una base de datos sin esquemas que solo requiere un nombre de tabla y una clave principal al crear la tabla.

Nombre de la tabla

Se utilizará para identificar su tabla.

TemperatureData

Entre 3 y 255 caracteres. Solo se pueden usar letras, números, guiones bajos (_) y puntos (.).

Clave de partición

La clave de partición forma parte de la clave principal de la tabla. Se trata de un valor hash que se utiliza para recuperar elementos de la tabla de escalabilidad y disponibilidad.

Año

Número

De 1 a 255 caracteres, distingue entre mayúsculas y minúsculas.

Clave de ordenación - opcional

Puede utilizar una clave de ordenación como segunda parte de la clave principal de una tabla. La clave de ordenación le permite ordenar o buscar los datos de la tabla.

Mes

Número

De 1 a 255 caracteres, distingue entre mayúsculas y minúsculas.

Configuración de la tabla

☒ Configuración predeterminada

La forma más rápida de crear su tabla. Puede modificar la mayor parte de la configuración después de crear la tabla. Para modificar esta configuración ahora, elija "Personalizar la configuración".

☐ Personalizar configuración

Utilice estas características avanzadas para personalizar la configuración.

Configuración básica Información

Descripción - Opcional

Memoria Información

La CPU asignada a la función es proporcional a la memoria configurada.

500 MB

Establezca la memoria en un valor entre 128 MB y 10240 MB.

Almacenamiento efímero Información

Puede configurar hasta 10 GB de almacenamiento efímero (/tmp) para la función. [Ver precios](#)

512 MB

Establezca el almacenamiento efímero (/tmp) entre 512 MB y 10240 MB.

SnapStart Información

Reduzca el tiempo de inicio haciendo que Lambda almacene en caché una instantánea de la función una vez esta se inicialice. Para evaluar instantáneas, consulte las [consideraciones de compatibilidad de SnapStart](#). Para las versiones ejecutables de Python y .NET, consulte la [documentación de SnapStart](#).

None

Tiempos de ejecución admitidos: .NET 8 (C#/.NET), Java 11, Java 17, Java 21, Python 3.12, Python 3.13.

Tiempo de espera

1 min 0 s

Pruebas del correcto funcionamiento de dynamo + lambda

Infraestructura para la Computación de Altas Prestaciones

Obtener el recuento de elementos en directo

Al elegir Comenzar análisis, realizará un análisis de DynamoDB para determinar el recuento de elementos más reciente. Este análisis podría consumir unidades de capacidad de lectura de tablas adicionales.

⚠ No se recomienda realizar esta acción en tablas muy grandes o en tablas que trabajan con tráfico de producción esencial. Puede pausar la acción en cualquier momento para evitar consumir capacidad de lectura adicional.

Recuento de elementos
89

Estado de análisis
Finalizado

Última actualización
enero 04, 2025 12:30:37

Analizar de nuevo

Cancelar

TemperatureData

Escanear o consultar elementos

Examen

Consulta

Seleccione una tabla o un índice
Tabla - TemperatureData

Seleccione la proyección de atributos
Todos los atributos

Filtros

Ejecutar Restablecer

Esta tabla tiene más elementos que recuperar. Para recuperar la siguiente página de elementos, elija Recuperar la siguiente página.

Elementos devueltos (50)

<input type="checkbox"/>	Año (Número)	Mes (Número)	DiferenciaTempMax	MaxDesviación	TempMaxMens...	TempMediaMensual
<input type="checkbox"/>	2017	3	0	0.40372043848...	17.32889501953...	17.056983947751906
<input type="checkbox"/>	2017	4	2.1354236602783203	0.62537944316...	19.46531867980...	18.872126897176106
<input type="checkbox"/>	2017	5	3.663116450078125	0.39043387446...	23.12843513488...	21.888142903645832
<input type="checkbox"/>	2017	6	5.100061416625977	0.41387853026...	28.22849655151...	26.325317764282225
<input type="checkbox"/>	2017	7	1.2870254516601562	0.32309707999...	29.51552200317...	28.274396419525146
<input type="checkbox"/>	2017	8	-2.083983685302734	0.52743977308...	27.43192863464...	27.431928634643555
<input type="checkbox"/>	2017	9	-0.3024061103515...	0.52055460214...	27.1295223236084	25.909506387247723
<input type="checkbox"/>	2017	10	-2.253877639770508	0.50646805763...	24.87564468581...	23.622084617614746
<input type="checkbox"/>	2017	11	-3.7722816467285156	0.49860823154...	21.10336303710...	17.21116790771484
<input type="checkbox"/>	2017	12	-6.296010971069356	0.90841162204...	11.80755206604...	11.608873128880991

Una vez terminado el pipeline, es momento de lanzar la aplicación. Puesto que hemos automatizado todo el proceso y explicado cada una de sus partes, consistirá simplemente en lanzar cada una de las plantillas en el siguiente orden.

En primer lugar añadimos la pila de red [proy-red](#), la cual creará toda la infraestructura. Es muy importante que el nombre que le pongamos a la pila coincida después con el que después importe la pila de aplicación [proy-app](#).

Pilas

red

2025-01-12 11:34:51 UTC+0100

CREATE_COMPLETE

c132287a335526418885935e1w829511480989

2025-01-04 10:10:00 UTC+0100

CREATE_COMPLETE

Marca temporal	ID lógico	Estado	Estado deta
2025-01-12 11:37:02 UTC+0100	red	CREATE_COMPLETE	-
2025-01-12 11:37:01 UTC+0100	proyPrivateSubnet1Route	CREATE_COMPLETE	-
2025-01-12 11:37:00 UTC+0100	proyPrivateSubnet1Route	CREATE_IN_PROGRES	-
2025-01-12 11:36:59 UTC+0100	proyPrivateSubnet1Route	CREATE_IN_PROGRES	-

Infraestructura para la Computación de Altas Prestaciones

Especificar los detalles de la pila

Proporcionar un nombre de pila

Nombre de la pila

El nombre de la pila debe tener entre 1 y 128 caracteres, comenzar con una letra y contener solo caracteres alfanuméricos. Recuento de caracteres: 10/128.

Parámetros

Los parámetros se definen en la plantilla y le permiten introducir valores personalizados al crear o actualizar una pila.

NetworkStackName

Name of an active CloudFormation stack that contains the networking resources, such as the VPC and subnet that will be used in this stack.

Observamos que automáticamente se aloja una imagen nueva en el ECR creado por la pila de red

▼ Private registry

Repositories

Summary

Images

Permissions

Lifecycle Policy

Repository tags

Container & Settings

Imágenes (1)

Eliminar Detalles Analizar Ver comandos de envío

☐

Etiqueta de imagen

▼

☐

latest

Tipo de artefacto

Image

Enviado a

12 de enero de 2025, 11:41:50 (UTC+01)

Tamaño (MB)

47.75

URI de imagen

Copiar URI

Resumir

sha256:b3595d155e587d...

Podríamos probar la aplicación localmente a través de SSM, para lo cual habría que utilizar la imagen y ejecutar un contenedor. Directamente realizaremos el autoescalado, siguiendo con la plantilla de Cluster [proy-cluster](#).

Debemos especificar los parámetros de la imagen (copiar URI de ECR) y el nombre de servicio deseado. Podemos ajustar el resto de valores, teniendo en cuenta que debe ser coherente. Es importante mantener el nombre de la pila de red al igual que en la pila que hicimos en la pila de aplicación. En nombres podemos escoger los que queramos. En la plantilla viene especificado un rango de 4-6 instancias EC2, las cuales se irán ajustando según necesidad

URI de la imagen del contenedor en ECR

ContainerPort

Puerto del contenedor

DesiredCount

Número deseado de tareas ECS

EnvironmentName

Nombre del entorno (ej. dev, prod)

NetworkStackName

Nombre del stack de red

ServiceName

Nombre del servicio ECS

Infraestructura para la Computación de Altas Prestaciones

Una vez finalizado, ya tenemos toda la estructura necesaria.

Podemos ver el mapeo de subredes y zonas de disponibilidad (Availability Zones) en AWS. Esto hace referencia a la configuración de red dentro de una VPC (Virtual Private Cloud) para mejorar la disponibilidad, redundancia y tolerancia a fallos de los recursos desplegados.

Mapeos

La selección de dos o más zonas de disponibilidad y las subredes correspondientes aumenta la tolerancia a errores de las aplicaciones.

Zona	Subred	Dirección IPv4	Dirección IPv4 privada	Dirección IPv6
us-east-1a (use1-az1)	subnet-096b2d6199d982d90	Asignado por AWS	Asignado desde el CIDR 10.0.1.0/24	No aplicable
us-east-1b (use1-az2)	subnet-0c88b40058f454637	Asignado por AWS	Asignado desde el CIDR 10.0.2.0/24	No aplicable

Comprobamos que los destinos estén en buen estado (Healthy) para la disponibilidad y el rendimiento de la aplicación, es importante verificar que especifica el puerto 5000.

Destinos registrados (2) [Info](#)

Mitigación de anomalías: **No aplicable**



Anular el registro

Registrar destinos

Los grupos de destinos enrutan las solicitudes a destinos individuales registrados mediante el protocolo y el número de puerto que especifique. Las comprobaciones de estado se realizan en todos los destinos registrados de acuerdo con la configuración de comprobación de estado del grupo de destinos. La detección de anomalías se aplica automáticamente a los grupos de destinos de HTTP/HTTPS con al menos 3 destinos en buen estado.

Filtrar destinos

< 1 > ⚙

<input type="checkbox"/>	dirección IP	Pue...	Zona	Estado	Detalles del estado	Sustitución administrativa	Detalle...	Resultado de la
<input type="checkbox"/>	10.0.3.236	5000	us-east-1a (...)	Healthy	-	No override	No overri...	Normal
<input type="checkbox"/>	10.0.3.119	5000	us-east-1a (...)	Healthy	-	No override	No overri...	Normal

Mostramos el grupo de Auto Scaling configurado para el clúster ECS, con una capacidad de 6 instancias activas (t2.micro), distribuidas entre zonas de disponibilidad us-east-1a y us-east-1b. Todas las instancias están en estado "InService" y "Healthy", gestionadas mediante una plantilla de lanzamiento. Esta configuración asegura alta disponibilidad y escalabilidad automática para las tareas ECS.

cluster-proyEcsAutoScalingGroup-3sKPNuRbScV5

cluster-proyEcsAutoScalingGroup-3sKPNuRbScV5 Descripción general de la capacidad

[Editar](#)

[arn:aws:autoscaling:us-east-1:829511480989:autoScalingGroup:8aecaecd-58bd-4aac-8c0b-3bb9adc4435f:autoScalingGroupName/cluster-proyEcsAutoScalingGroup-3sKPNuRbScV5](#)

Capacidad deseada	Límites de escalamiento (Mín. - Máx.)	Tipo de capacidad deseado	Estado
6	4 - 6	Unidades (número de instancias)	-

Fecha de creación
Sun Jan 05 2025 19:22:37 GMT+0100 (hora estándar de Europa central)

Detalles Integraciones - nueva Escalado automático **Administración de instancias** Actualización de instancias Actividad Monitoreo

Instancias (6)



Acciones

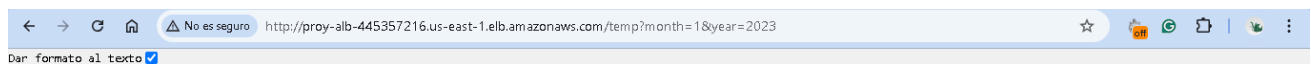
Filtrar las instancias

< 1 > ⚙

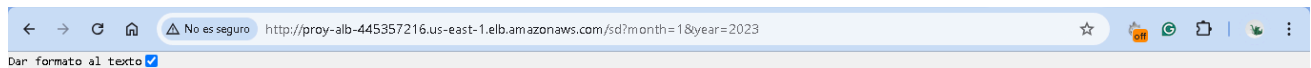
<input type="checkbox"/>	ID de instancia	Ciclo de v...	Tipo de i...	Capacida...	Plantilla ...	Zona de ...	Estado	Protegido ...
<input type="checkbox"/>	i-02f2d0c357dbba174	InService	t2.micro	-	asgLaunchTempla	us-east-1a	Healthy	Escalado des...
<input type="checkbox"/>	i-09ee35c7ad9c30ff0	InService	t2.micro	-	asgLaunchTempla	us-east-1b	Healthy	Escalado des...
<input type="checkbox"/>	i-0acb3a47662cd7b19	InService	t2.micro	-	asgLaunchTempla	us-east-1b	Healthy	Escalado des...
<input type="checkbox"/>	i-0ae786a02b515db3b	InService	t2.micro	-	asgLaunchTempla	us-east-1a	Healthy	Escalado des...
<input type="checkbox"/>	i-0cc3f83180656cd3a	InService	t2.micro	-	asgLaunchTempla	us-east-1b	Healthy	Escalado des...

Infraestructura para la Computación de Altas Prestaciones

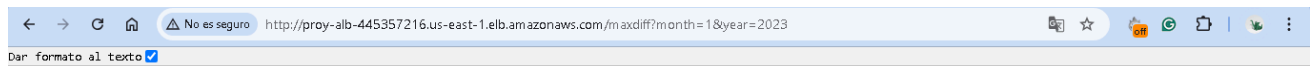
Con toda la infraestructura configurada correctamente, el siguiente paso es verificar que el balanceador de carga funcione adecuadamente. Para ello, se copia la URL proporcionada por el balanceador y se realiza una consulta utilizando la dirección en un navegador. Si todo funciona correctamente se puede observar la respuesta esperada generada por la aplicación Flask, confirmando que el balanceador está distribuyendo el tráfico de manera efectiva hacia las instancias registradas.



```
{
  "Año": 2023,
  "Mes": 1,
  "TemperaturaMedia": 12.920413306619
}
```

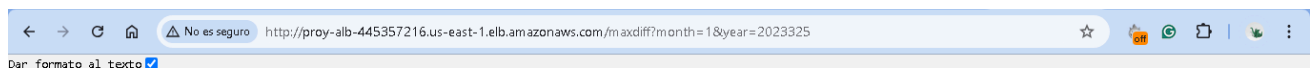


```
{
  "Año": 2023,
  "DesviacionTípica": 0.256263043536181,
  "Mes": 1
}
```

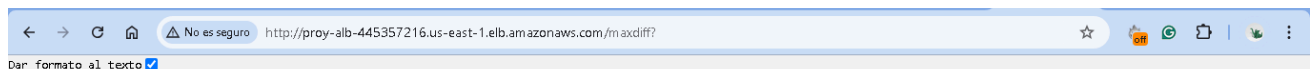


```
{
  "Año": 2023,
  "Maxdiff": -0.964802128823616,
  "Mes": 1
}
```

Ejemplo tratamiento de errores.



```
{
  "message": "No se encontraron resultados para la fecha proporcionada"
}
```



```
{
  "error": "Los parámetros 'month' y 'year' son obligatorios"
}
```

La dirección del balanceador de carga es: `proy-ALB-445357216.us-east-1.elb.amazonaws.com`

Resumen de recursos y servicios desplegados

Crea cuantas filas como sea necesario. Debes incluir todos los recursos que hacen posible que tu despliegue funcione correctamente.

Identificador de recurso	Nombre de recurso	Tipo de recurso en AWS	Comentario (opcional)
arn:aws:elasticloadbalancing:us-east-1:829511480989:loadbalancer/app/proy-ALB/ada44a681304d211	proy-ALB	Balanceador de carga	
829511480989.dkr.ecr.us-east-1.amazonaws.com/proy-repositorio	proy-repositorio	ECR	Hub para alojar imagen Docker
vpc-0cb43a15fba1225bf	proy-MyVPC	VPC	
subnet-0991f0db77424d7	proy-PrivateSubnet1	Subred Privada	
subnet-03ce70155932d7b5b	proy-PrivateSubnet2	Subred Privada	
subnet-0df158aeed0119ab4	proy-PublicSubnet1	Subred Pública	

Infraestructura para la Computación de Altas Prestaciones

subnet-0d4b0b537725ecc37	proy-PublicSubnet2	Subred Pública	
rtb-05c34abdcd7066bf6	proy-PublicRouteTable	Tabla de enrutamiento	
rtb-03f9e80a2b3a42551	proy-PrivateRouteTable	Tabla de enrutamiento	
igw-0bc6c4926d17bbe1b	proy-MyInternetGateway	Internet Gateway	Conexión de las subredes públicas a internet
nat-03b99e94162d1b4fa	proy-NatGateway	NAT	NAT para conexión a internet de las redes privadas
arn:aws:s3:::landingzone-virginia	landingzone-virginia	S3	Ingesta de datos crudos
arn:aws:s3:::summaryfiles-virginia	summaryfiles-virginia	S3	Almacenamiento persistente de datos filtrados de forma mensual y semanal
arn:aws:lambda:us-east-1:829511480989:function:filterfiles	filterfiles	lambda function	Filtrado inicial de datos
arn:aws:lambda:us-east-1:829511480989:function:glue_auto	glue_auto	lambda function	automatización desencadenamiento glue

Infraestructura para la Computación de Altas Prestaciones

arn:aws:lambda:us-east-1:829511480989:function:csv_to_dynamo	csv_to_dynamo	lambda function	Transformación de los datos analizados a dynamoDB
arn:aws:sns:us-east-1:829511480989:SD_LIMIT	SD_LIMIT	SNS Subscription	Suscripción desviación típica superior a 0.5
arn:aws:ecs:us-east-1:829511480989:cluster/proy-Cluster	Proy-Cluster	Cluster ECS	
arn:aws:ecs:us-east-1:829511480989:service/proy-Cluster/proy-servicio	proy-servicio	Servicio	
sg-0300eab3cc9f605df	proy-ALB Security Group	Grupos de seguridad	
sg-0c6b606a6968c424a	proy-PrivateFlask Security Group	Grupos de seguridad	
sg-00727eb5d5616ac79	proy-Docker Security Group	Grupos de seguridad	
sg-073706641283b6720	proy-VpcEndpointSecurityGroup	Grupos de seguridad	
arn:aws:ecs:us-east-1:829511480989:task-definition/proy-Task:5	proy-Task:5	Definición de tarea	

Infraestructura para la Computación de Altas Prestaciones

arn:aws:autoscaling:us-east-1:829511480989:autoScalingGroup:d3679b48-8e7a-4bed-9c52-5024011de824:autoScalingGroupName/cluster-proyEcsAutoScalingGroup-oKzrwY9YAfKF	cluster-proyEcsAutoScalingGroup-oKzrwY9YAfKF	Auto-Scaling	
--	--	--------------	--