Universidad de San Andrés - Maestría en Ciencia de Datos

Trabajo práctico final

Programación Avanzada para Grandes Volúmenes de datos

Profesores:

- Agustín Mosteiro
- Matías Dinota

Alumnos:

- Federico Gonzalez Pietranera
- Ezequiel Kinigsberg
- Rafael Zambrano

Link al repositorio: https://github.com/ezekini/pagvd

Introducción

El objetivo de este trabajo práctico fue desarrollar y poner en producción un sistema de recomendación de productos en el ámbito de la industria AdTech. Este sistema se compone de dos partes fundamentales: un pipeline de procesamiento de datos para generar las recomendaciones y una API para servir dichas recomendaciones. Para implementar y desplegar estas soluciones, utilizamos varias herramientas y servicios de AWS, como EC2, ECS, ECR, RDS y S3.

Desarrollo del Proyecto

Pipeline de datos: para generar las recomendaciones diarias, desarrollamos un pipeline usando Apache Airflow que corre en una instancia de AWS EC2. Este pipeline realiza varias tareas:

1. Filtrado de datos: Lee los datos crudos de S3 y elimina a aquellos advertisers inactivos.
2. Cálculo de TopCTR: Calcula los productos con mejor click-through-rate para cada advertiser activo.
3. Cálculo de TopProduct: Determina los productos más vistos en la web del cliente.
4. Escritura en la BBDD: Escribe los resultados en una base de datos en AWS RDS. Además mantiene un historial de siete días en BD, eliminando datos antigüos.

El flujo de trabajo del pipeline se orquestó usando Airflow, configurado con un LocalExecutor y una base de datos PostgreSQL para manejar las tareas en paralelo.

API de Recomendaciones

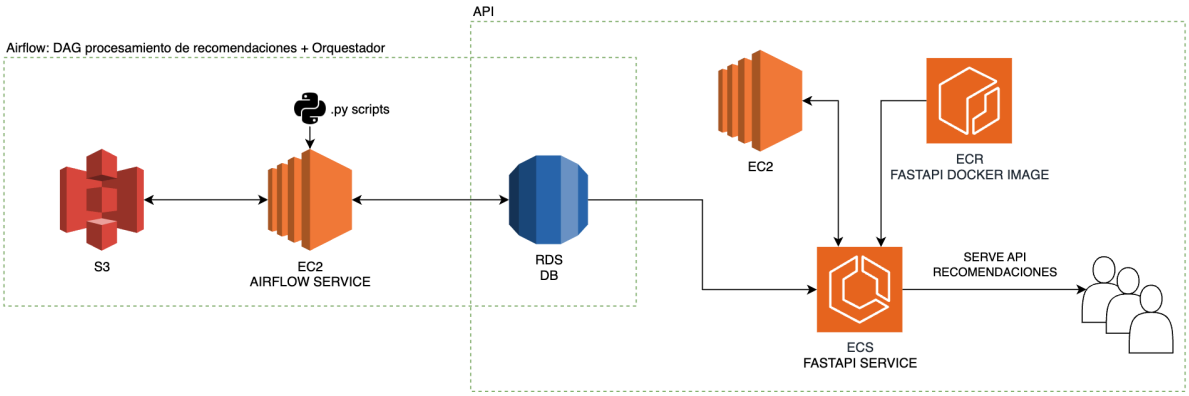
La API, implementada en FastAPI, está dockerizada y desplegada en AWS ECS. Provee tres endpoints principales:

1. */recommendations/<ADV>/<Modelo>*: Devuelve las recomendaciones del día para un advertiser específico y modelo.
2. */stats/*: Ofrece un resumen de estadísticas sobre las recomendaciones. En particular devuelve para cada advertiser y fecha el porcentaje de productos recomendados coincidentes entre ambos modelos.
3. */history/<ADV>/*: Muestra el historial de recomendaciones para un advertiser en los últimos siete días.

Base de Datos

La base de datos PostgreSQL en AWS RDS es el núcleo de almacenamiento donde se guardan las recomendaciones generadas por el pipeline y de donde la API lee para servir los datos en tiempo real.

Arquitectura de servicios



Dificultades Encontradas

Durante el desarrollo, nos enfrentamos a varios desafíos:

- Deploy del Contenedor Docker en AWS ECR y ECS: si bien crear y construir el contenedor Docker no fue complejo, desplegarlo en el servicio de Elastic Container Registry (ECR) presentó problemas. Configurar correctamente el servicio de Elastic Container Service (ECS) para que utilizara esta imagen y el hardware adecuado requirió bastante investigación y pruebas.
- Configuración de Airflow en EC2: Ejecutar Airflow en un entorno virtual dentro de una instancia de EC2 fue complicado. En primera instancia pip no estaba dentro del entorno por lo que no podíamos instalar de la forma en que estamos acostumbrados hasta resolver ese problema. Tuvimos que probar diversos paquetes y versiones debido a múltiples problemas: el servicio de Airflow no se levantaba, el scheduler se caía, no podíamos acceder a la UI, etc. Por ejemplo, Airflow instaló una versión caduca de SQLAlchemy que no era compatible con la versión de pandas utilizada en el pipeline.
- Otro inconveniente encontrado se dio respecto al concepto de *catchup* en Airflow. Cada vez que se levantaba el servicio, el DAG se ejecutaba todos los días donde no corrió, incluso estando en teoría desactivada esta función.
- El manejo de datos respecto a las fechas para compararlas y cumplir con la consigna también requirió especial atención

Conclusión

Este proyecto nos permitió comprender en profundidad la integración y despliegue de aplicaciones utilizando servicios de AWS. Aprendimos a manejar la orquestación de tareas con Airflow, el despliegue de contenedores con Docker y ECS y la configuración de bases de datos en RDS como así también a generar una API sencilla con algunos endpoints.