

1- Objetivo

O objetivo deste trabalho é desenvolver um sistema de monitoramento e análise de dados de clientes de uma plataforma de delivery, integrando um modelo de Inteligência Artificial (IA) para segmentação de comportamento de consumo.

O sistema foi implementado e executado em uma instância EC2 da AWS, utilizando contêineres Docker para isolar os serviços de banco de dados, geração de dados sintéticos e execução do modelo.

2- Arquitetura do Sistema

Serviço	Tecnologia	Função
Banco de Dados	PostgreSQL	Armazena informações de clientes, pedidos e restaurantes
Gerador de Dados	Python + Faker	Cria dados fictícios de clientes e pedidos para simulação
Dashboard / API IA	Next.js + Prisma	Exibe os resultados e disponibiliza endpoints REST para consumo externo

3 – EC2 (AWS)

Instalando Docker na AWS:

```
admin@ip-172-31-20-56:~$ curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
admin@ip-172-31-20-56:~$ echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] \
https://download.docker.com/linux/debian \
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
admin@ip-172-31-20-56:~$ sudo apt update -y
Get:1 file:/etc/apt/mirrors/debian.list Mirrorlist [38 B]
Get:2 file:/etc/apt/mirrors/debian-security.list Mirrorlist [47 B]
Get:7 https://download.docker.com/linux/debian trixie InRelease [32.5 kB]
Get:8 https://download.docker.com/linux/debian trixie/stable amd64 Packages [16.5 kB]
Hit:3 https://cdn-aws.deb.debian.org/debian trixie InRelease
Hit:4 https://cdn-aws.deb.debian.org/debian trixie-updates InRelease
Hit:5 https://cdn-aws.deb.debian.org/debian trixie-backports InRelease
Hit:6 https://cdn-aws.deb.debian.org/debian-security trixie-security InRelease
Fetched 49.0 kB in 1s (81.0 kB/s)
All packages are up to date.
admin@ip-172-31-20-56:~$ sudo apt install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin -y
Installing:
containerd.io docker-buildx-plugin docker-ce docker-ce-cli docker-compose-plugin
```

Baixa e adiciona a chave GPG oficial do Docker — usada para validar pacotes.

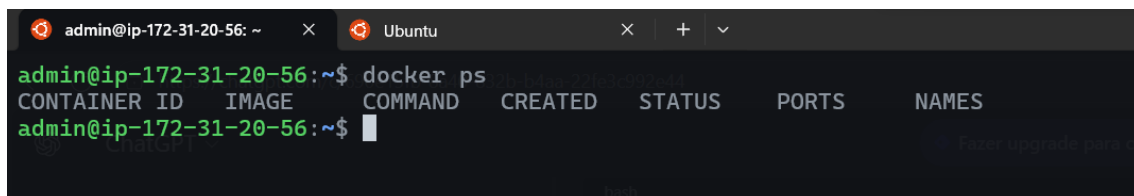
Adiciona o repositório oficial do Docker à lista de fontes do sistema Debian.

Atualiza os pacotes disponíveis (sudo apt update).

Instala os componentes principais:

- docker-ce: Engine do Docker.
- docker-ce-cli: Interface de linha de comando do Docker.
- containerd.io: Gerenciador de contêineres em baixo nível.
- docker-buildx-plugin: Plugin para builds otimizados.
- docker-compose-plugin: Novo Compose integrado (comando docker compose).

Verificando instalação do Docker com PS:

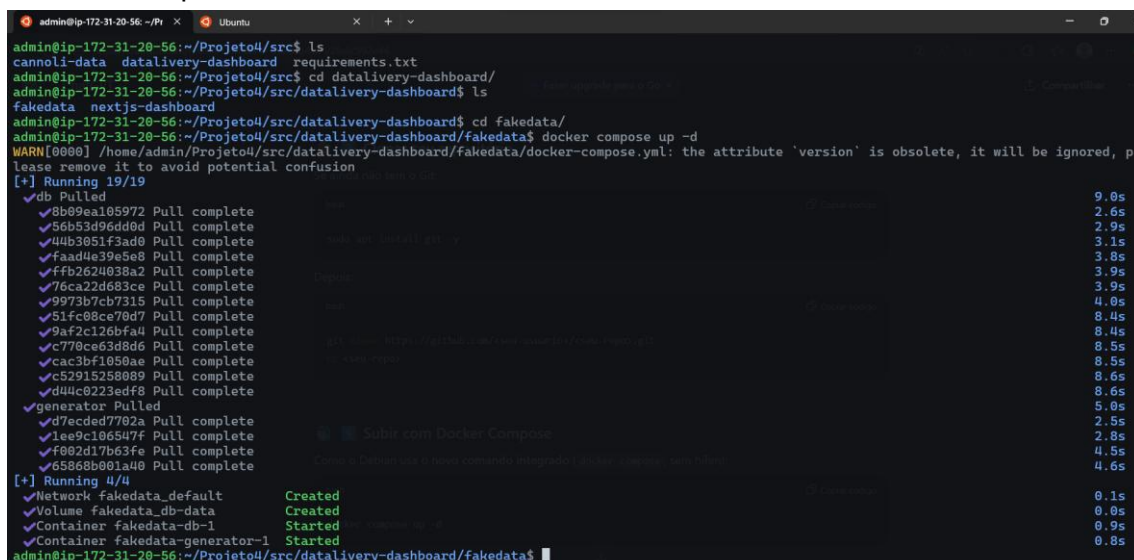


```
admin@ip-172-31-20-56: ~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
admin@ip-172-31-20-56: ~$
```

O que faz:

- Lista todos os contêineres em execução no momento.
- Como o retorno está vazio, significa que nenhum contêiner está ativo ainda, mas o Docker está funcionando corretamente.

Docker Compose:



```
admin@ip-172-31-20-56: ~/Projeto4/src$ ls
cannoli-data  datalivery-dashboard  requirements.txt
admin@ip-172-31-20-56: ~/Projeto4/src$ cd datalivery-dashboard/
admin@ip-172-31-20-56: ~/Projeto4/src/datalivery-dashboard$ ls
fakedata  nextjs-dashboard
admin@ip-172-31-20-56: ~/Projeto4/src/datalivery-dashboard$ cd fakedata/
admin@ip-172-31-20-56: ~/Projeto4/src/datalivery-dashboard/fakedata$ docker compose up -d
WARN[0000] /home/admin/Projeto4/src/datalivery-dashboard/fakedata/docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
[*] Running 19/19
✓db Pulled
  8b09ea105972 Pull complete
  56b53d96dd0d Pull complete
  44b3051f3ad0 Pull complete
  faad4e39e5e8 Pull complete
  ffb2624038a2 Pull complete
  76ca22d683ce Pull complete
  9973b7cb7315 Pull complete
  51fc08ce70d7 Pull complete
  9af2c126bfa4 Pull complete
  c770ec63d8d6 Pull complete
  cac3bf1050ae Pull complete
  c52915258089 Pull complete
  d44c0223edf8 Pull complete
✓generator Pulled
  d7ecded7702a Pull complete
  1ee9c106547f Pull complete
  f002d17b63fe Pull complete
  65868b001a40 Pull complete
[*] Running 4/4
✓Network fakedata_default Created
✓Volume fakedata_db-data Created
✓Container fakedata-db-1 Started
✓Container fakedata-generator-1 Started
admin@ip-172-31-20-56: ~/Projeto4/src/datalivery-dashboard/fakedata$
```

O que o Compose fez automaticamente:

1. Criou a rede Docker para comunicação entre serviços.
2. Criou um volume persistente para armazenar dados do PostgreSQL.
3. Subiu o banco de dados (db) e o script gerador (generator).
4. Populou o banco com cerca de 12.000 pedidos fictícios.

4 – Verificando containers na AWS três dias após cria-los

Docker ps -a pra verificar os containers ativos

```
PowerShell
admin@ip-172-31-20-56: ~
12:10 satiro ~ $ ssh -i ~/.ssh/debianAWS.pem admin@18.188.53.121
Linux ip-172-31-20-56 6.12.41+deb13-cloud-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.12.41-1 (2025-08-12) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Nov 7 16:09:55 2025 from 177.98.102.185
admin@ip-172-31-20-56:~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
ff40b672a8af   python:3.11-slim "bash -lc 'pip insta..." 2 days ago    Exited (0) 2 days ago
57177d025bdc   postgres:15     "docker-entrypoint.s..." 2 days ago    Up 2 days    0.0.0.0:5432->
admin@ip-172-31-20-56:~$
```

Docker stats pra verificar o uso de processamento

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PI
57177d025bdc	fakedata-db-1	0.00%	33.08MiB / 1.891GiB	1.71%	13.4MB / 3.94MB	8.19kB / 471MB	6

Docker logs pra ver os registros do container

```
admin@ip-172-31-20-56:~$ docker logs fakedata-db-1
The files belonging to this database system will be owned by user "postgres".
This user must also own the server process.

The database cluster will be initialized with locale "en_US.utf8".
The default database encoding has accordingly been set to "UTF8".
The default text search configuration will be set to "english".

Data page checksums are disabled.

fixing permissions on existing directory /var/lib/postgresql/data ... ok
creating subdirectories ... ok
selecting dynamic shared memory implementation ... posix
selecting default max_connections ... 100
selecting default shared_buffers ... 128MB
selecting default time zone ... Etc/UTC
creating configuration files ... ok
running bootstrap script ... ok
performing post-bootstrap initialization ... ok
syncing data to disk ... ok

Success. You can now start the database server using:

pg_ctl -D /var/lib/postgresql/data -l logfile start
```

5 – Conclusão

O projeto Datalivery demonstra a integração entre infraestrutura em nuvem, persistência de dados e modelos de IA aplicados a problemas reais de negócio.

O uso de K-Means possibilitou uma análise automatizada de comportamento de clientes, identificando padrões de fidelidade e valor para a empresa.

Além disso, a execução em ambiente Dockerizado na AWS garante reprodutibilidade, escalabilidade e fácil integração com sistemas corporativos.