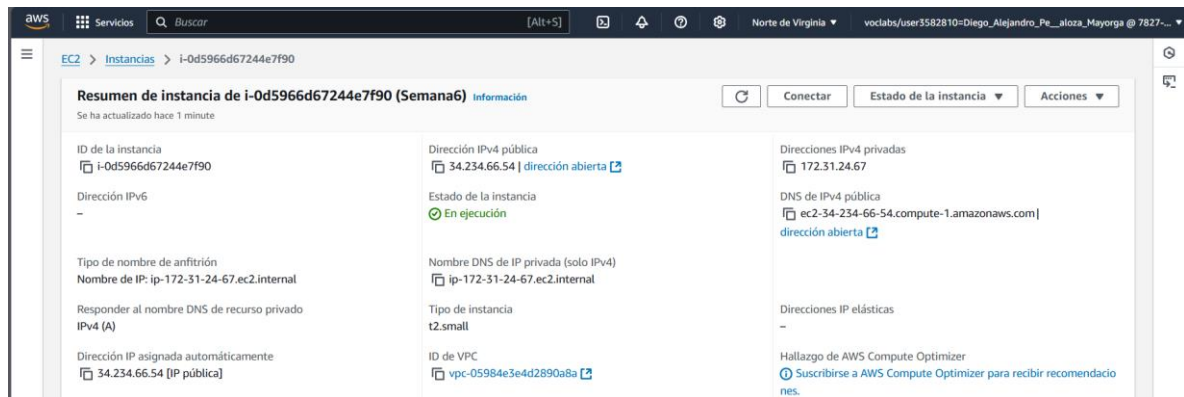


Entrega Semana 6

Realizado por: Diego Peñaloza

PARTE 1: Despliegue de contenedores

En la consola de EC2 lance una instancia t2.small, Ubuntu server con 20 GB de disco. Incluya un pantallazo de la consola de AWS EC2 con la máquina en ejecución en su reporte. Su usuario de AWS y las IPs privada y pública deben estar visible en el pantallazo.



Para verificar su instalación, descargue, construya y ejecute la imagen hello-world sudo docker run hello-world Copie y explique la salida en pantalla en su reporte.

```
ubuntu@ip-172-31-24-67:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:305243c734571da2d100c8c8b3c3167a098cab6049c9a5b066b6021a60fcb966
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

Explore el archivo Dockerfile y en su reporte explique brevemente el paso a paso de la configuración definida en este archivo. Para esto tenga presente que FROM determina la imagen base que se usa como sistema operativo y aplicaciones iniciales. RUN ejecuta comandos al interior del contenedor. COPY copia archivos del sistema hospedador (la máquina virtual) al contenedor. ENV permite definir variables de entorno. EXPOSE abre un puerto del contenedor. CMD es el comando que se ejecuta al lanzar el contenedor.

En el archivo Dockerfile inicialmente se define la imagen base la cual será Python:3.10

```
FROM python:3.10
```

Paso seguido en RUN se crea un nuevo usuario sin contraseña que ejecuta la aplicación. Además, se establece opt/bankchurn-api como el directorio de trabajo.

```
# Crear usuario que ejecuta la app
RUN adduser --disabled-password --gecos '' api-user

# Definir directorio de trabajo
WORKDIR /opt/bankchurn-api
```

Se copia el contenido de la carpeta bankchurn-api en el directorio de trabajo previamente definido. Luego, se actualiza pip y se instalan las dependencias definidas en el archivo requirements.txt.

```
# Instalar dependencias
ADD ./bankchurn-api /opt/bankchurn-api/
RUN pip install --upgrade pip
RUN pip install -r /opt/bankchurn-api/requirements.txt
```

A partir del comando chmod +x se garantiza que el archivo run.sh pueda ejecutarse correctamente. Además, se asegura que solo el usuario api-user tenga control sobre los archivos de la aplicación.

```
# Hacer el directorio de trabajo ejecutable
RUN chmod +x /opt/bankchurn-api/run.sh
# Cambiar propiedad de la carpeta a api-user
RUN chown -R api-user:api-user ./
```

Se configura que el único usuario en modificar la aplicación sea api-user. Además, se abre el puerto 8001 para ejecutar la API. Por último, con el comando CMD se especifica que cuando el dockerfile se ejecute este debe ejecutar el archivo run.sh en una terminal Bash, por consiguiente dando inicio a la API.

```
USER api-user
# Puerto a exponer para la api
EXPOSE 8001

# Comandos a ejecutar al correr el contenedor
CMD ["bash", "./run.sh"]
```

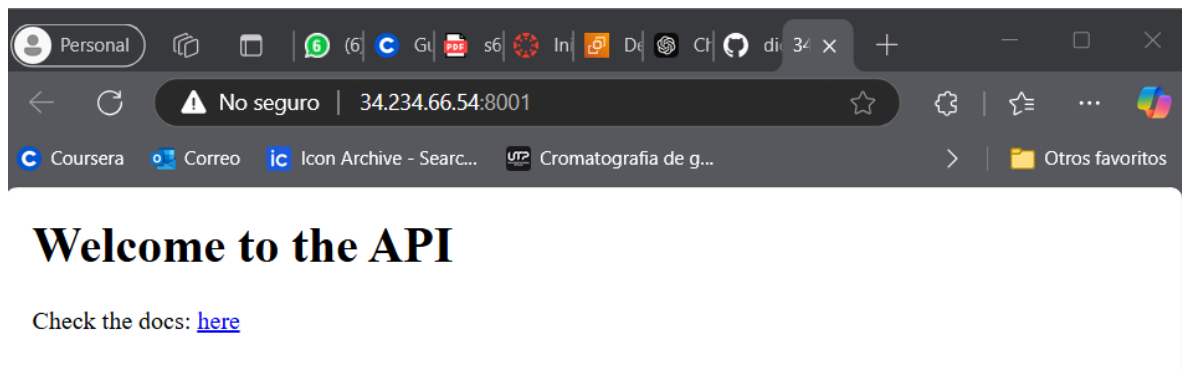
Liste las imágenes de docker con el comando `sudo docker images` Debe contar con la imagen recién creada de bankchurn-api y la hello-world creada anteriormente. Incluya un pantallazo de la salida en su reporte. Su IP privada debe ser visible.

```
ubuntu@ip-172-31-24-67:~/DSA_S6_Docker$ sudo docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
bankchurn-api       latest          81a8a44ee770   About a minute ago  1.56GB
hello-world         latest         d2c94e258dcb   18 months ago   13.3kB
```

Ahora ejecute un contenedor usando la imagen creada con el comando `sudo docker run -p 8001:8001 -it -e PORT=8001 bankchurn-api` Incluya un pantallazo de la salida de este comando en su reporte. La IP privada debe ser visible.

```
ubuntu@ip-172-31-24-67:~/DSA_S6_Docker$ sudo docker run -p 8001:8001 -it -e PORT=8001 bankchurn-api
INFO: Started server process [7]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: Uvicorn running on http://0.0.0.0:8001 (Press CTRL+C to quit)
2024-11-15 01:53:39.423 | INFO | uvicorn.protocols.http.h11_impl:send:477 - 186.86.32.61:6854 - "GET / HTTP/1.1" 200
2024-11-15 01:53:39.582 | INFO | uvicorn.protocols.http.h11_impl:send:477 - 186.86.32.61:6854 - "GET /favicon.ico HTTP/1.1" 404
```

Copie la IP pública de su máquina y en un navegador local visite la página IP:8001. Allí debe aparecer la API de bankchurn en ejecución. Incluya un pantallazo del navegador en su reporte. La dirección (IP pública) debe ser visible.

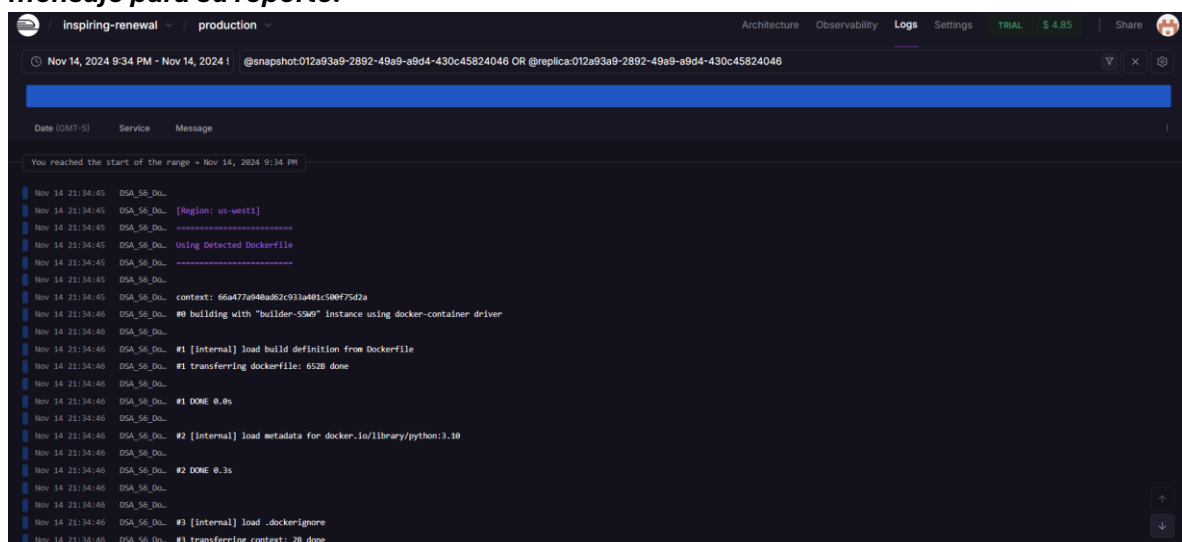


Parte 2: despliegue con Railway

Con el enlace realizado, estamos listos para lanzar la aplicación en Railway con el comando `railway up --detach`. Tome un pantallazo de la salida de este comando e inclúyala en su reporte.

```
ubuntu@ip-172-31-24-67:~/DSA_S6_Docker$ railway up --detach
Indexed
Compressed [=====] 100%
Uploaded
Build Logs: https://railway.app/project/0c8ec7b9-9c9c-4414-a228-38080f5e9bd7/service/7fa70b63-7c35-4eaa-8ba7-ef5488df6f65?id=2e22fdbc-d6be-44e2-a570-74432b2d1ff0&
```

Regrese al sitio de Railway, en su proyecto y servicio. De click en la pestaña Deployments, allí verá que el despliegue se está realizando. De click en viewlogs e identifique que al principio se indique que se está usando el Dockerfile encontrado: Using detected Dockerfile. Tome un pantallazo de los logs de despliegue con este mensaje para su reporte.

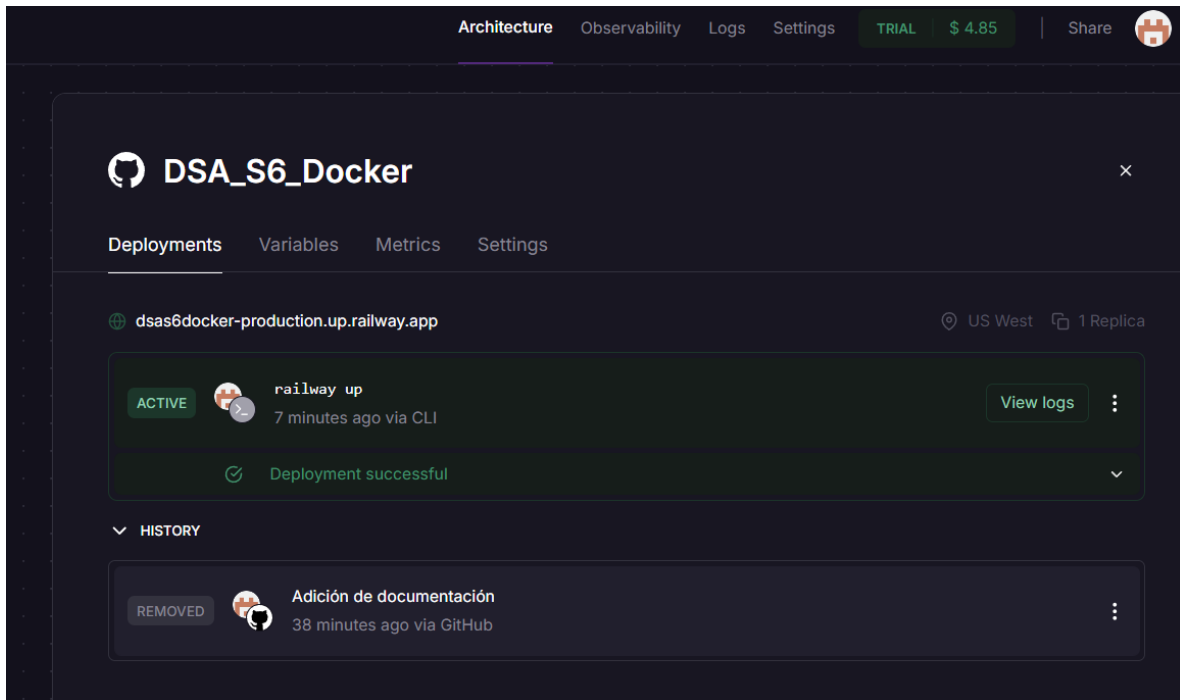


```
Nov 14 21:34:46 DSA_S6_Do... #3 DONE 0.0s
Nov 14 21:34:46 DSA_S6_Do...
Nov 14 21:34:46 DSA_S6_Do...
Nov 14 21:34:46 DSA_S6_Do... #4 [1/8] FROM docker.io/library/python:3.10@sha256:941b0bfddb17d809fd1f457acbf55dfca014e3e0e3d592b1c9070491681bc02
Nov 14 21:34:46 DSA_S6_Do... #4 resolve docker.io/library/python:3.10@sha256:941b0bfddb17d809fd1f457acbf55dfca014e3e0e3d592b1c9070491681bc02 done
Nov 14 21:34:46 DSA_S6_Do... #4 DONE 0.0s
Nov 14 21:34:46 DSA_S6_Do...
Nov 14 21:34:46 DSA_S6_Do... #5 [internal] load build context
Nov 14 21:34:46 DSA_S6_Do... #5 transferring context: 6.75MB 0.0s done
Nov 14 21:34:46 DSA_S6_Do... #5 DONE 0.0s
Nov 14 21:34:46 DSA_S6_Do...
Nov 14 21:34:46 DSA_S6_Do... #6 [2/8] RUN adduser --disabled-password --gecos '' api-user
Nov 14 21:34:46 DSA_S6_Do... #6 CACHED
Nov 14 21:34:46 DSA_S6_Do...
Nov 14 21:34:46 DSA_S6_Do... #7 [6/8] RUN pip install -r /opt/bankchurn-api/requirements.txt
Nov 14 21:34:46 DSA_S6_Do... #7 CACHED
Nov 14 21:34:46 DSA_S6_Do...
Nov 14 21:34:46 DSA_S6_Do... #8 [4/8] ADD ./bankchurn-api /opt/bankchurn-api/
Nov 14 21:34:46 DSA_S6_Do... #8 CACHED
Nov 14 21:34:46 DSA_S6_Do...
Nov 14 21:34:46 DSA_S6_Do... #9 [7/8] RUN chmod +x /opt/bankchurn-api/run.sh
Nov 14 21:34:46 DSA_S6_Do... #9 CACHED
Nov 14 21:34:46 DSA_S6_Do...
Nov 14 21:34:46 DSA_S6_Do... #11 CACHED
Nov 14 21:34:46 DSA_S6_Do...
Nov 14 21:34:46 DSA_S6_Do... #12 [8/8] RUN chown -R api-user:api-user ./
Nov 14 21:34:46 DSA_S6_Do... #12 CACHED
Nov 14 21:34:46 DSA_S6_Do...
Nov 14 21:34:46 DSA_S6_Do... #13 exporting to image
Nov 14 21:34:46 DSA_S6_Do... #13 exporting layers 0.0s done
Nov 14 21:34:46 DSA_S6_Do... #13 exporting manifest sha256:14bfbb2fc2d73dd881a82f392613a0ce053e09f3dc56a1d7c1dda59f423002c8 done
Nov 14 21:34:46 DSA_S6_Do... #13 exporting config sha256:f17ff17212a0e54429dd1dee56d2889df981561ac5f4da254b5e9738fab7ffec done
Nov 14 21:34:46 DSA_S6_Do... #13 exporting attestation manifest sha256:a8c9412becfcbc734226ba2f6a3629498cf0119e4d411f98a6bd80d0d08d6a9c done
Nov 14 21:34:46 DSA_S6_Do... #13 exporting manifest list sha256:5af6f4ff5f2a1b495785726231433935c58f74ddf6d4c9bf8c0fa80864ebaa3d
Nov 14 21:34:46 DSA_S6_Do...
Nov 14 21:34:46 DSA_S6_Do... #13 ...
Nov 14 21:34:46 DSA_S6_Do...
Nov 14 21:34:46 DSA_S6_Do... #14 [auth] sharing credentials for us-west1.registry.rlyw.net
Nov 14 21:34:46 DSA_S6_Do... #14 DONE 0.0s
Nov 14 21:34:46 DSA_S6_Do...
Nov 14 21:34:46 DSA_S6_Do... #13 exporting to image
Nov 14 21:34:46 DSA_S6_Do... #13 exporting manifest list sha256:5af6f4ff5f2a1b495785726231433935c58f74ddf6d4c9bf8c0fa80864ebaa3d done
Nov 14 21:34:46 DSA_S6_Do... #13 pushing layers
Nov 14 21:34:46 DSA_S6_Do...
You reached the end of the range → Nov 14, 2024 9:34 PM
```

```
inspiring-renewal production Architecture Observability Logs Settings TRIAL $ 4.85 Share
14 minutes ago @deployment:2e22fcb-d8be-44e2-a570-74432b2dfff0 -@replica:012a93a9-2892-49a9-a9d4-430c45824046
Date (GMT-5) Service Message
You reached the start of the range → Nov 14, 2024 9:34 PM
Nov 14 21:35:05 DSA_S6_Do... Starting Container
Nov 14 21:35:07 DSA_S6_Do... INFO: Started server process [2]
Nov 14 21:35:07 DSA_S6_Do... INFO: Waiting for application startup.
Nov 14 21:35:07 DSA_S6_Do... INFO: Application startup complete.
Nov 14 21:35:07 DSA_S6_Do... INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
Nov 14 21:39:18 DSA_S6_Do... 2024-11-15 02:39:14.008 | INFO | uvicorn.protocols.http.h11_impl:send:477 - 100.64.0.2:24202 - "GET / HTTP/1.1" 200
Nov 14 21:39:18 DSA_S6_Do... 2024-11-15 02:39:14.301 | INFO | uvicorn.protocols.http.h11_impl:send:477 - 100.64.0.2:24202 - "GET /favicon.ico HTTP/1.1" 404
Nov 14 21:39:18 DSA_S6_Do... 2024-11-15 02:39:15.559 | INFO | uvicorn.protocols.http.h11_impl:send:477 - 100.64.0.2:24202 - "GET /docs HTTP/1.1" 200
Nov 14 21:39:18 DSA_S6_Do... 2024-11-15 02:39:16.154 | INFO | uvicorn.protocols.http.h11_impl:send:477 - 100.64.0.2:24202 - "GET /api/v1/openapi.json HTTP/1.1" 200
Nov 14 21:42:48 DSA_S6_Do... 2024-11-15 02:42:41.545 | INFO | uvicorn.protocols.http.h11_impl:send:477 - 100.64.0.2:27452 - "GET / HTTP/1.1" 200
Nov 14 21:42:48 DSA_S6_Do... 2024-11-15 02:42:42.937 | INFO | uvicorn.protocols.http.h11_impl:send:477 - 100.64.0.2:27452 - "GET /docs HTTP/1.1" 200
Nov 14 21:42:48 DSA_S6_Do... 2024-11-15 02:42:43.229 | INFO | uvicorn.protocols.http.h11_impl:send:477 - 100.64.0.2:27452 - "GET /api/v1/openapi.json HTTP/1.1" 200
```

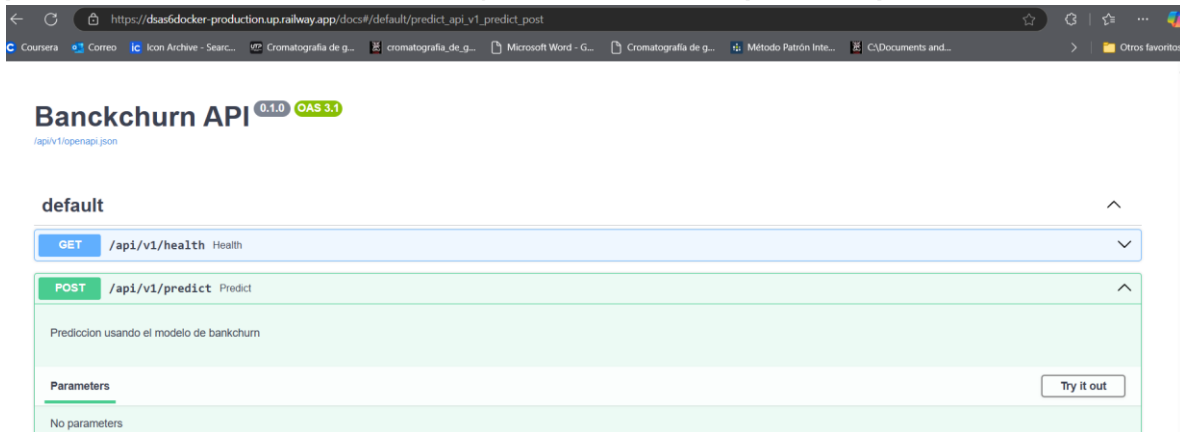
Cuando termine de desplegar, queda en estado Active. Regrese a la ventana anterior (click en la x de la esquina superior derecha). Click en la pestaña Settings. En la sección

Networking y Public Networking, click en Generate Domain. Al generar el dominio tome un pantallazo del dominio generado para su reporte.



<https://dsas6docker-production.up.railway.app/>

Click en here para ir a los docs. Expanda la ruta POST /api/v1/predict y tome un pantallazo donde sea visible la URL y las rutas de la API para su reporte.



Parameters

Try it out

No parameters

Request body required

application/json

Example Value | Schema

```
{
  "inputs": [
    {
      "Customer_Age": 57,
      "Gender": "M",
      "Dependent_count": 4,
      "Education_Level": "Graduate",
      "Marital_Status": "Single",
      "Income_Category": "$120K +",
      "Card_Category": "Blue",
      "Months_on_book": 52,
      "Total_Relationship_Count": 2,
      "Months_Inactive_12_mon": 3,
      "Contacts_Count_12_mon": 0,
      "Credit_Limit": 25000,
      "Total_Revolving_Bal": 0,
      "Avg_Open_To_Buy": 25000,
      "Total_Amt_Chng_Q4_Q1": 0.712,
      "Total_Trans_Amt": 7794,
      "Total_Trans_Ct": 94,
      "Total_Ct_Chng_Q4_Q1": 0.843,
      "Avg_Utilization_Ratio": 0
    }
  ]
}
```

Responses

Code

Description

Links

200

Successful Response

No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{
  "errors": "string",
  "version": "string",
  "predictions": [
    0
  ]
}
```

422

Validation Error

No links

Media type

application/json

Example Value | Schema

```
{
  "detail": [
    {
      "loc": [
        "string",
        0
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```