

Manual de instalación del tablero

Objetivo:

Este manual de instalación comprende un instructivo detallado que tiene como objetivo guiarle en el proceso de despliegue de la API y del tablero a través del uso de EC2 de AWS y el servicio de PaaS de Railway con Docker. Este documento es una adaptación de la Guía de Trabajo Práctico. Semana 6. Taller 1 Docker Railway Parte 2.

Despliegue API:

1. En su equipo local primero descargue la carpeta [API_final](https://github.com/juankquintana/prediccion_salarios/tree/main/DespliegueFinal/API_final), del repositorio https://github.com/juankquintana/prediccion_salarios/tree/main/DespliegueFinal/API_final
Cree su propio repositorio en github con ella. (También puede hacer un fork de nuestro repositorio lo que genera una copia del mismo). El repositorio creado será utilizado al momento del despliegue luego de haber creado la instancia EC2 y de hacer el link con Railway.
2. En la consola de EC2 lance una instancia t2.small, Ubuntu server con 20 GB de disco. Genere o utilice una llave PEM, descárguela en su máquina local. Esta le servirá como autenticación al acceder a la instancia.
3. Para conectarse a la instancia, en una terminal emita el comando:

```
ssh -i /path/to/llave.pem ubuntu@IP
```

donde /path/to/ se refiere a la ubicación del archivo llave.pem que descargó, e IP es la dirección IP pública de la instancia EC2 que lanzó.

Si prefiere, en la terminal puede navegar a la ubicación del archivo llave.pem y emitir el comando:

```
ssh -i llave.pem ubuntu@IP
```

4. Para desplegar la API emplearemos Railway App. Si no tiene una cuenta, creela en <https://railway.app/dashboard>. Podrá acceder a un plan con 5 dólares de créditos, que deben ser suficientes para ésta actividad. Note que puede usar su cuenta de GitHub para crear la cuenta Railway.
5. Para emplear Railway necesitará el manejador de paquetes npm en su máquina virtual.

En la máquina virtual actualice el listado de paquetes:

```
sudo apt update
```

6. Luego instale npm en su versión estándar para Ubuntu:

```
sudo apt install npm
```

7. Ahora queremos actualizar npm y node para contar con versiones recientes, necesarias para Railway. Para esto instale nvm con el comando:

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash
```

8. A continuación corra el comando:

```
source ~/.bashrc
```

9. Revise la versión de nvm y las versiones remotas disponibles de node:

```
nvm --version
```

```
nvm ls-remote
```

10. Instale la última versión LTS (long-term support). En el ejemplo se utiliza la 20.9:

```
nvm install v20.9.0
```

11. Revise la versión de node y npm:

```
node --version
```

```
npm --version
```

12. Ahora sí podemos instalar railway usando npm:

```
npm i -g @railway/cli
```

13. Revise la versión de railway:

```
railway --version
```

14. Loguéese a railway con el comando:

```
railway login --browserless
```

Este comando genera una URL, copie esta URL en un navegador (Chrome, Firefox, etc.) donde tenga abierta su cuenta Railway (equipo local). Verifique el acceso en el navegador y revise en la terminal de la máquina (virtual) que quedó logueado.

15. En su navegador asegúrese de tener abiertas sus cuentas de GitHub y Railway: si ya ha usado Railway con Github vaya al paso 16, si es la primera vez usando Railway con Github, en el navegador vaya a su cuenta de Railway y enlace su cuenta con Github.
- click en New project,
 - seleccione Deploy from GitHub repo,
 - click en Configure GitHub App.

Seleccione la cuenta de Github donde instalará la aplicación Railway.

Permita el acceso a todos los repositorios. Confirme usando su método preferido de verificación.

16. Con su cuenta ya configurada y enlazada con GitHub, cree un nuevo proyecto vacío:
- Click en New project,
 - Seleccione Empty project.
17. En el proyecto creado agregue un servicio vacío:
- click en Add a service,
 - seleccione Empty service
18. Justo arriba del servicio aparecerá una ventana para aplicar los cambios en el proyecto, que corresponden a la creación del servicio. Dé clic en 'Deploy'.
19. En el proyecto encontrará en la parte inferior una caja con el mensaje:
"Set up your project locally".

Al hacer click allí encontrará un paso de conexión con la siguiente forma:

`railway link -p IDPROYECTO`

donde IDPROYECTO es un código de identificación del proyecto. Copie este ID. Cierre la ventana de diálogo.

20. De regreso en la terminal de su máquina virtual clone el repositorio creado al inicio en la parte de descarga de la API:

`git clone https://github.com/username/RepositorioAPI.git`

21. Ingrese a la carpeta del repositorio, liste los contenidos de la carpeta

`ls -la`

Verifique que tiene la carpetas salarios-api-abc, así como los archivos .git, y Dockerfile.

22. Desde esta carpeta enlace el proyecto y el servicio de Railway con link obtenido:

`railway link -p proyectoid`

23. Con el enlace realizado, estamos listos para lanzar la api en Railway con el comando

`railway up --detach`

24. Regrese al sitio de Railway, en su proyecto y servicio. De click en la pestaña Deployments -> View Logs, allí verá que el despliegue se está realizando.

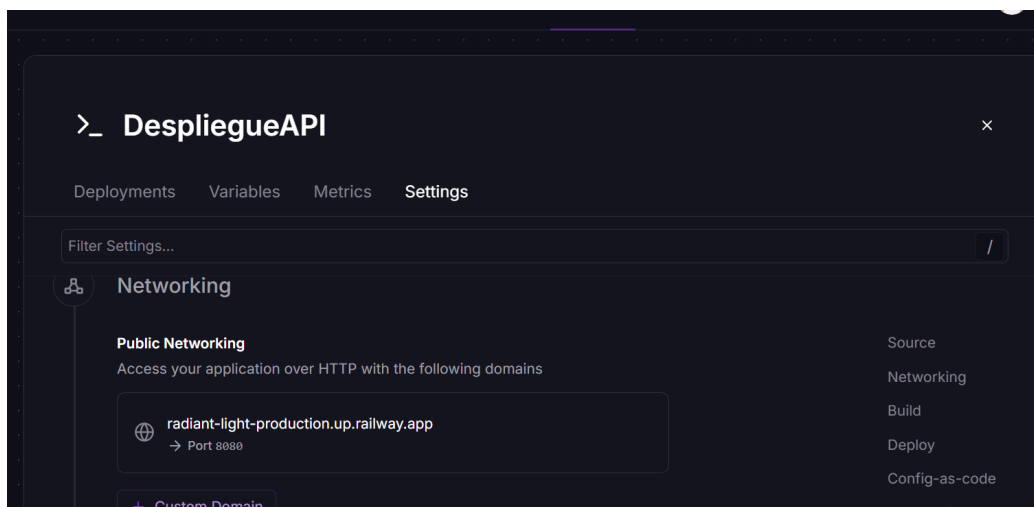
Cuando termine de desplegar, quedará en estado Active.

25. Regrese a la ventana anterior (click en la x de la esquina superior derecha).
Seleccione la pestaña Settings.

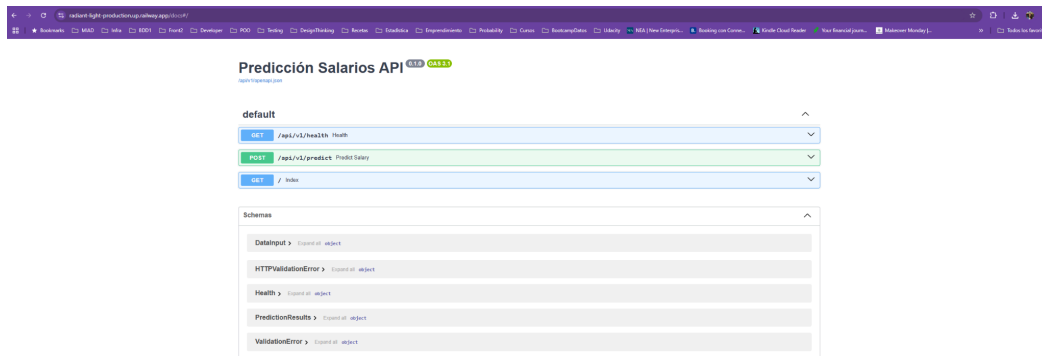
En la sección Networking y Public Networking, click en Generate Domain.

Si solicita incluir puerto indique 8080

Note que se ha creado una URL. Esta corresponde al API_URL que deberá utilizar en la parte del despliegue del Dashboard.



26. De click en la URL o cópiela y péguela en su navegador. Note que se despliega la API.
Guárdela para utilizarla más adelante.



Despliegue Tablero

Ahora usaremos otra máquina virtual para desplegar allí el tablero empleando EC2 y Railway. Tenga en cuenta que es imprescindible mantener encendida la máquina virtual utilizada en la parte anterior de despliegue de la API.

27. Antes de comenzar con el despliegue en su equipo local primero descargue la carpeta **DASH_final**, del repositorio https://github.com/juankquintana/prediccion_salarios/tree/main/DespliegueFinal/DASH_final

Cree su propio repositorio en github con ella. (También puede hacer un fork de nuestro repositorio lo que genera una copia del mismo).

El repositorio creado será utilizado más adelante.

Ahora es necesario repetir los puntos ya realizados anteriormente para generar una nueva instancia EC2 y un nuevo proyecto y deployment en Railway para utilizar en el despliegue del tablero:

28. En la consola de EC2 lance una instancia t2.small, Ubuntu server con 20 GB de disco. Genere o utilice una llave PEM, descárguela en su máquina local. Esta le servirá como autenticación al acceder a la instancia.

Para conectarse a la instancia, en una terminal emita el comando:

```
ssh -i /path/to/llave.pem ubuntu@IP
```

donde /path/to/ se refiere a la ubicación del archivo llave.pem que descargó, e IP es la dirección IP pública de la instancia EC2 que lanzó.

Si prefiere, en la terminal puede navegar a la ubicación del archivo llave.pem y emitir el comando:

```
ssh -i llave.pem ubuntu@IP
```

29. Para emplear Railway necesitará el manejador de paquetes npm en su máquina virtual.

En la máquina virtual actualice el listado de paquetes:

```
sudo apt update
```

30. Luego instale npm en su versión estándar para Ubuntu:

```
sudo apt install npm
```

31. Ahora queremos actualizar npm y node para contar con versiones recientes, necesarias para Railway. Para esto instale nvm con el comando:

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash
```

32. A continuación corra el comando:

```
source ~/.bashrc
```

33. Revise la versión de nvm y las versiones remotas disponibles de node:

```
nvm --version
```

```
nvm ls-remote
```

34. Instale la última versión LTS (long-term support) de node, el ejemplo corresponde a la 20.9:

```
nvm install v20.9.0
```

35. Revise la versión de node y npm:

```
node --version
```

```
npm --version
```

36. Ahora sí podemos instalar railway usando npm:

```
npm i -g @railway/cli
```

37. Revise la versión de railway:

`railway --version`

38. Loguéese a railway con el comando:

`railway login --browserless`

Este comando genera una URL, copie esta URL en un navegador (Chrome, Firefox, etc.) donde tenga abierta su cuenta Railway (equipo local). Verifique el acceso en el navegador y revise en la terminal de la máquina (virtual) que quedó logueado.

39. En su navegador asegúrese de tener abiertas sus cuentas de GitHub y Railway:

40. En Railway cree un nuevo proyecto vacío:

- a. Click en New project,
- b. Seleccione Empty project.

41. En el proyecto creado agregue un servicio vacío:

- a. click en Add a service,
- b. seleccione Empty service

Ahora tendrá dos proyectos con servicios asociados, uno donde estará corriendo la API y otro donde realizará el despliegue del tablero.

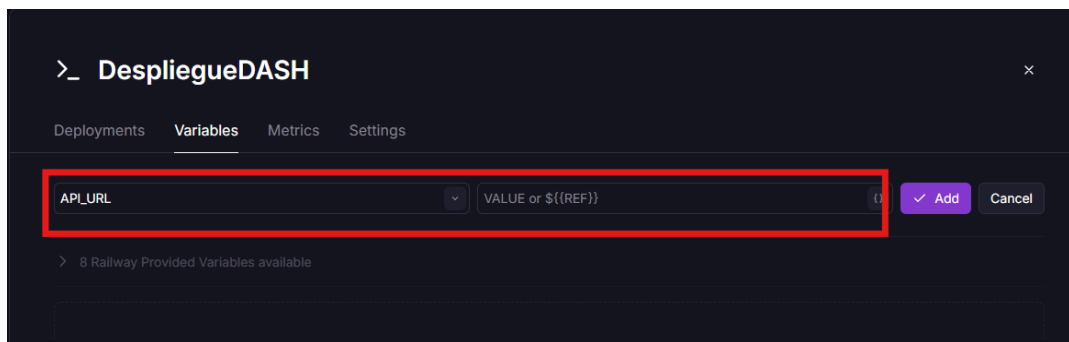
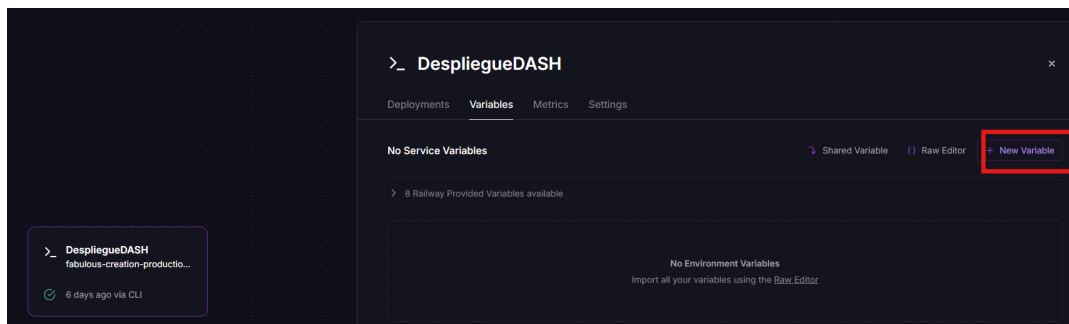
42. Justo arriba del servicio aparecerá una ventana para aplicar los cambios en el proyecto, que corresponden a la creación del servicio. Dé clic en 'Deploy'.

43. En el proyecto encontrará en la parte inferior una caja con un mensaje "Set up your project locally". Al hacer click allí encontrará un paso de conexión con la siguiente forma:

`railway link -p IDPROYECTO`

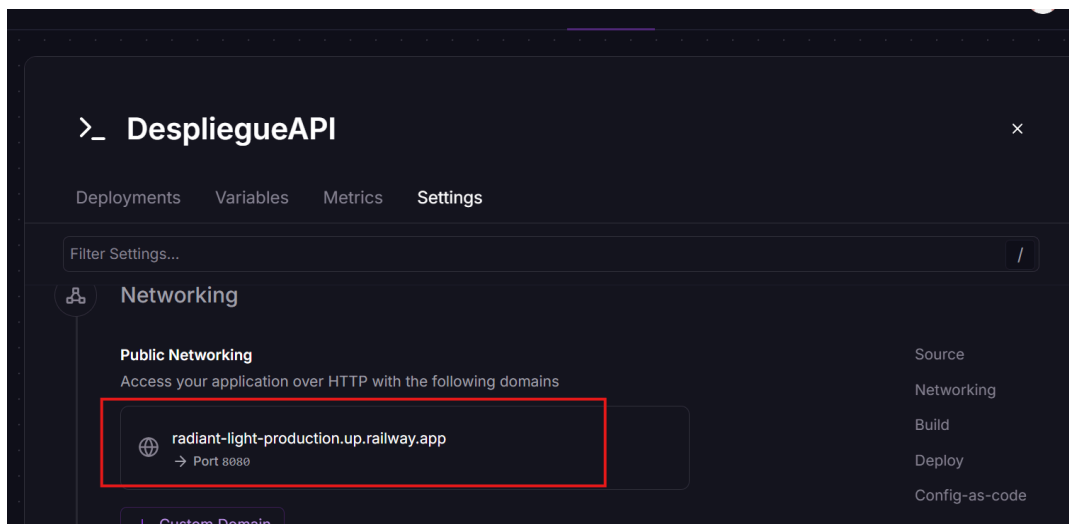
donde IDPROYECTO es un código de identificación del proyecto. Copie este ID. Cierre la ventana de diálogo.

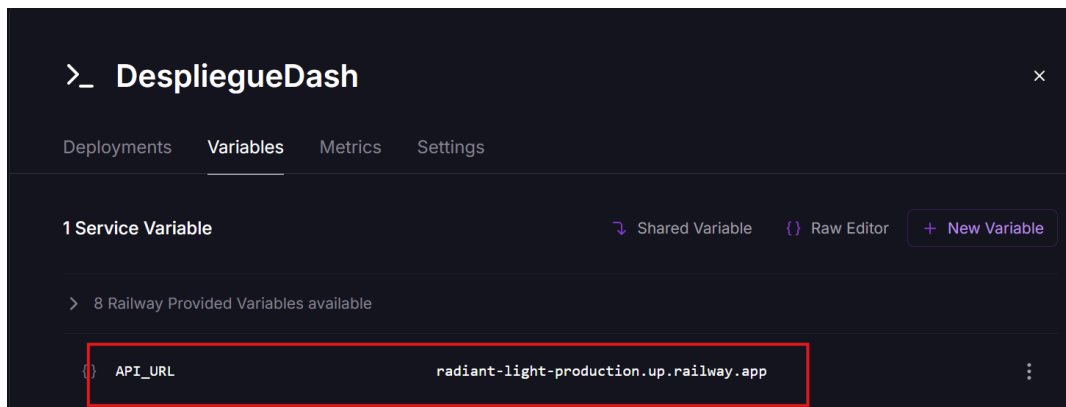
44. En Railway en el Servicio que acaba de crear para desplegar el dash es necesario que agregue una variable de entorno que debe llamar: **API_URL** y como valor debe ingresar el dominio público obtenido en obtenido en los puntos 25 y 26 de este instructivo:



En el ejemplo incluido en este instructivo la API_URL es radiant-light-production.up.railway.app

Nuevamente arriba del servicio aparecerá una ventana para aplicar los cambios en el proyecto, que corresponden a la creación de la nueva variable. Dé clic en 'Deploy'.





Cuando usted genere su propio proyecto este dominio será distinto.

45. De regreso en la terminal de su máquina virtual clone el repositorio creado al inicio en la parte de descarga del tablero:

```
git clone https://github.com/username/RepositorioDash.git
```

46. Ingrese a la carpeta del repositorio, liste los contenidos de la carpeta

```
ls -la
```

47. Verifique que tiene la carpeta `app`, así como los archivos `.git` y `Dockerfile`.

48. Desde esta carpeta enlace el proyecto y el servicio de Railway con link obtenido en Railway:

```
railway link -p proyectoid
```

49. Con el enlace realizado, estamos listos para lanzar el tablero en Railway con el comando

```
railway up --detach
```

50. Regrese al sitio de Railway, en su proyecto y servicio. De click en la pestaña Deployments -> View Logs, allí verá que el despliegue se está realizando.

Cuando termine de desplegar, quedará en estado Active.

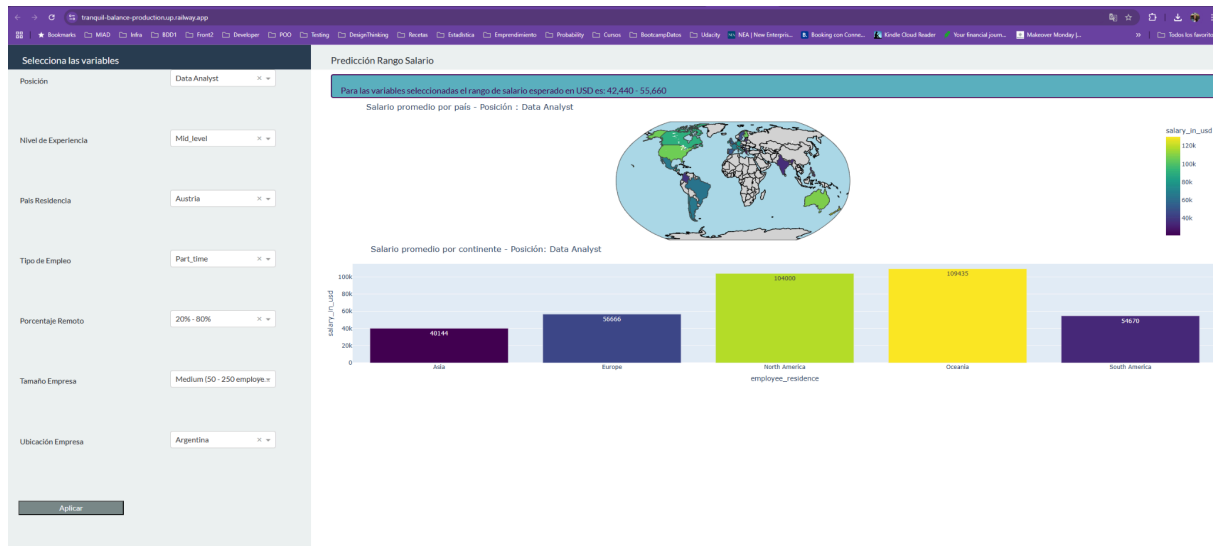
51. Regrese a la ventana anterior (click en la x de la esquina superior derecha). Click en la pestaña Settings.

En la sección Networking y Public Networking, click en Generate Domain.

Si solicita incluir puerto indique 8080

Note que se ha creado una URL.

52. De click en la URL o cópiela y péguela en su navegador. Note que se despliega el Tablero y puede interactuar con él y ver la interacción en el panel Deployments/View logs de cada servicio (API y DASH):



```
>_ DespliegueDash / ca034cb Nov 21, 2024 10:07 PM x
ACTIVE tranquil-balance-production.up.railway.app
Details Build Logs Deploy Logs HTTP Logs Filter logs using "", (), AND, OR, -
Date (GMT-3) Message
You reached the start of the range -> Nov 21, 2024 10:07 PM
Nov 21 22:08:26 Starting Container
Nov 21 22:08:28 [2024-11-22 01:08:27 +0000] [2] [INFO] Starting gunicorn 21.2.0
Nov 21 22:08:28 [2024-11-22 01:08:27 +0000] [2] [INFO] Listening at: http://0.0.0.0:8080 (2)
Nov 21 22:08:28 [2024-11-22 01:08:27 +0000] [2] [INFO] Using worker: sync
Nov 21 22:08:28 [2024-11-22 01:08:27 +0000] [3] [INFO] Booting worker with pid: 3
Nov 21 22:08:58 2024-11-22 01:08:50.226 | INFO | app:update_output:209 - Payload sent to API: {'job_title': 'Data Analyst', 'experience_level': 'Mid_level', 'employee_country': 'Austria', 'company_country': 'Argentina'}
Nov 21 22:08:58 2024-11-22 01:08:50.347 | INFO | app:update_output:236 - Prediction range calculated: 42,440 - 55,660
Nov 22 15:06:54 2024-11-22 18:06:47.516 | INFO | app:update_output:209 - Payload sent to API: {'job_title': 'Data Analyst', 'experience_level': 'Mid_level', 'employee_country': 'Austria', 'company_country': 'Argentina'}
Nov 22 15:06:54 2024-11-22 18:06:49.643 | INFO | app:update_output:236 - Prediction range calculated: 42,440 - 55,660
```

DespliegueAPI / 164a35d Nov 21, 2024 8:09 PM

ACTIVE radiant-light-production.up.railway.app

Details Build Logs Deploy Logs HTTP Logs Filter logs using "", (), AND, OR, -

Date (GMT-3)	Message
	/api/v1/predict HTTP/1.1" 405
Nov 21 22:05:06	2024-11-22 01:05:06.465 INFO uvicorn.protocols.http.h11_impl:send:477 - 100.64.0.3:20282 - "GET /api/v1/predict HTTP/1.1" 405
Nov 21 22:05:26	2024-11-22 01:05:22.459 INFO app.api:predict:45 - Received input data: job_title='Software Engineer' experience_level='Mid_level' employee_country='United States' company_country='United States'
Nov 21 22:05:26	2024-11-22 01:05:22.462 INFO app.api:predict:60 - Prediction result: 90010.20617027677
Nov 21 22:05:26	2024-11-22 01:05:22.463 INFO uvicorn.protocols.http.h11_impl:send:477 - 100.64.0.7:44032 - "POST /api/v1/predict HTTP/1.1" 200
Nov 21 22:06:16	2024-11-22 01:06:12.876 INFO app.api:predict:45 - Received input data: job_title='Data Analyst' experience_level='Mid_level' employee_country='United States' company_country='United States'
Nov 21 22:06:16	2024-11-22 01:06:12.877 INFO app.api:predict:60 - Prediction result: 52617.331831317555
Nov 21 22:06:16	2024-11-22 01:06:12.878 INFO uvicorn.protocols.http.h11_impl:send:477 - 100.64.0.12:33412 - "POST /api/v1/predict HTTP/1.1" 200
Nov 21 22:08:57	2024-11-22 01:08:50.331 INFO app.api:predict:45 - Received input data: job_title='Data Analyst' experience_level='Mid_level' employee_country='Austria' company_country='Argentina'
Nov 21 22:08:57	2024-11-22 01:08:50.333 INFO app.api:predict:60 - Prediction result: 52617.331831317555
Nov 21 22:08:57	2024-11-22 01:08:50.333 INFO uvicorn.protocols.http.h11_impl:send:477 - 100.64.0.13:36956 - "POST /api/v1/predict HTTP/1.1" 200
Nov 22 15:06:54	2024-11-22 18:06:48.175 INFO app.api:predict:45 - Received input data: job_title='Data Analyst' experience_level='Mid_level' employee_country='Austria' company_country='Argentina'
Nov 22 15:06:54	2024-11-22 18:06:49.572 INFO app.api:predict:60 - Prediction result: 52617.331831317555
Nov 22 15:06:54	2024-11-22 18:06:49.596 INFO uvicorn.protocols.http.h11_impl:send:477 - 100.64.0.6:45414 - "POST /api/v1/predict HTTP/1.1" 200

En el documento [Manual de usuario del tablero](#) puede encontrar el detalle de uso del tablero.

53. Finalmente, para terminar por completo el despliegue de la API y el tablero, y dejar de usar recursos computacionales, debe:

- Terminar la ejecución del servicio en Railway. Seleccione el servicio y allí el menú Deployments. Seleccione el despliegue en verde (ejecución), click en los 3 puntos y seleccione Remove.
- Terminar el uso de las instancias en AWS. Para ello selecciónelas en la consola de AWS y terminelas.

BIBLIOGRAFÍA

Guia de Trabajo Práctico. Semana 6. Taller 1 Docker Railway Parte 2.

Varios Autores

Universidad de los Andes

Maestría Inteligencia Analítica de Datos. Despliegue de Soluciones.