

# Lancer l'API et l'Interface Web avec Docker

Pour exécuter le backend (API FastAPI) et le frontend (interface Streamlit) via Docker, suivez ces étapes :

---

## 1 Pré-requis

- Installer **Docker** sur votre machine :  
[Télécharger Docker](#)
- 

## 2 Ouvrir un terminal

- Sur Windows : PowerShell ou Terminal WSL
  - Sur macOS/Linux : Terminal classique
- 

## 3 Construire les images Docker

Depuis le dossier contenant votre `docker-compose.yml` :

```
In [ ]: docker-compose build --no-cache
```

## 4 Lancer les conteneurs en arrière-plan

```
In [ ]: docker-compose up -d
```

- `-d` = détaché, les conteneurs tournent en arrière-plan.
- Vérifier que tout est lancé :

```
In [ ]: docker-compose ps
```

```

MP_DATA> streamlit run frontend.py

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://192.168.1.22:8501

PS C:\MP_DATA> uvicorn backend:app --reload
INFO: Will watch for changes in these directories: ['C:\MP_DATA']
INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO: Started reloader process [16100] using StatReload
C:\MP_DATA>
[22:00:06] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-08cbc033d8d4aae1-1\xgboost\xgboost-ci-windows\src\data\..\common\error_msg.h:80: I
f you are loading a serialized model (like pickle in Python, RDS in R) or
configuration generated by an older version of XGBoost, please export the model by calling
'Booster.save_model' from that version first, then load it back in current version. See:
https://xgboost.readthedocs.io/en/stable/tutorials/saving_model.html
for more details about differences between saving model and serializing.

warnings.warn(msg, UserWarning)
INFO: Started server process [22896]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: 127.0.0.1:25077 - "POST /predict HTTP/1.1" 200 OK

```

## 5 Accéder à l'interface web

- Ouvrir un navigateur et taper :

In [ ]: `http://localhost:8501/`

- L'interface Streamlit devrait s'afficher.

Deploy

## NBA Career Predictor

Entrez les statistiques du joueur pour prédire la durée probable de sa carrière

Nombre de matchs joués (GP)

50

-

+

Minutes jouées par match (MIN)

26,00

-

+

Points par match (PTS)

10,00

-

+

Tirs réussis (FGM)

4,00

-

+

Tirs tentés (FGA)

7,00

-

+

Passes décisives (AST)

3,70

Interceptions (STL)

0,90

Contres (BLK)

5,00

Ballons perdus (TOV)

1,00

Prédire carrière

Probabilité carrière  $\geq$  5 ans : 0.70

Prédiction finale : Longue carrière

- L'API FastAPI est accessible sur :

In [ ]: `http://localhost:8000/docs`

## Arrêter les conteneurs

In [ ]: `docker-compose down`

In [ ]: