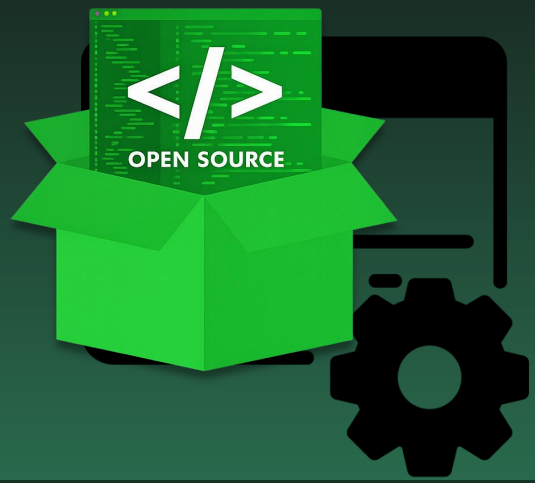
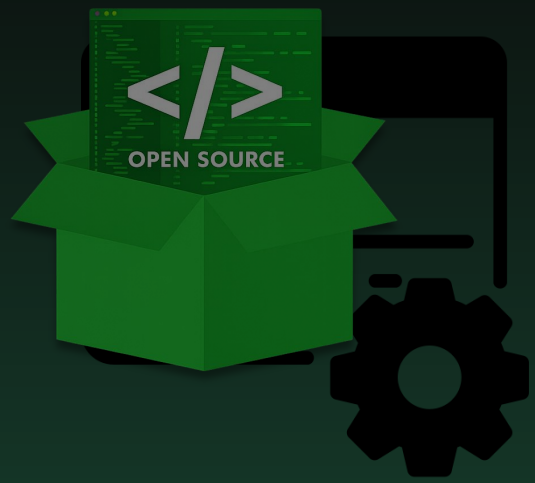

supabase

supabase

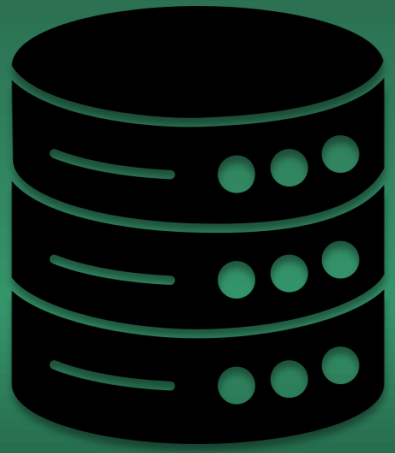


open-source
backend
platform

supabase

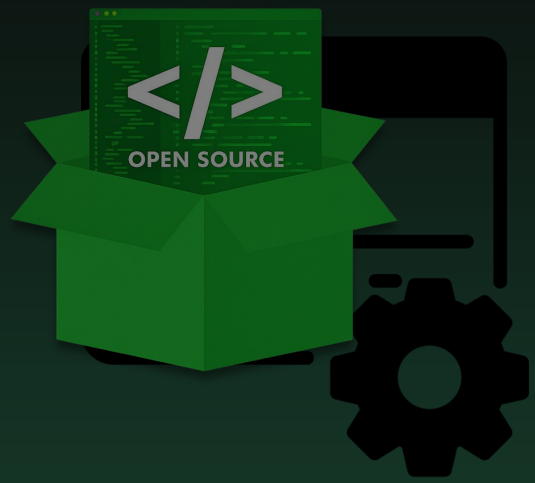


open-source
backend
platform

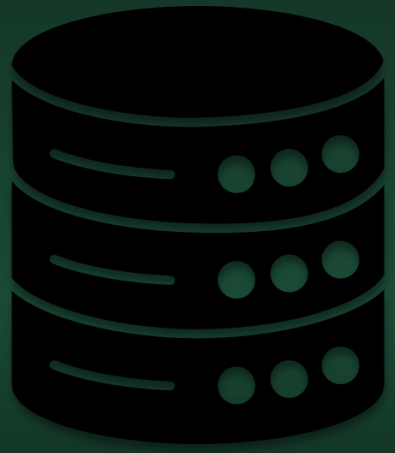


Build and manage
databases

supabase



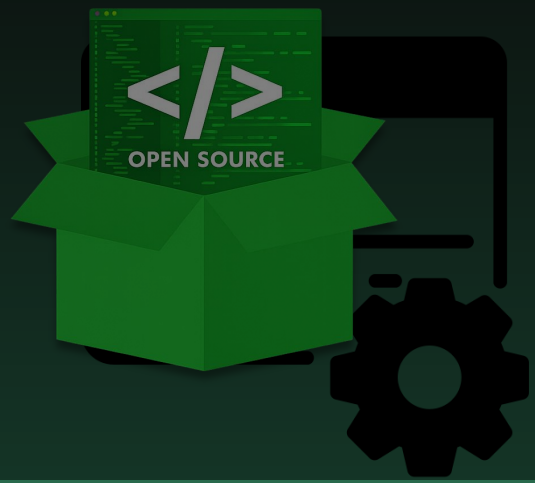
open-source backend platform



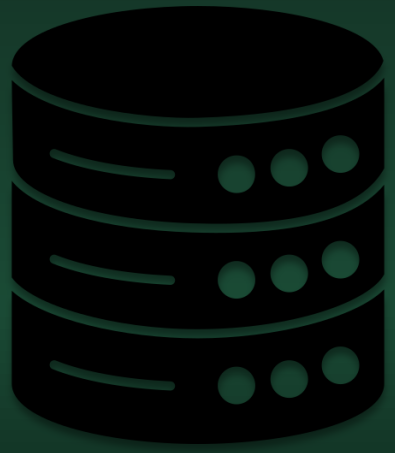
Build and manage databases

Handle user authentication

supabase



open-source backend platform

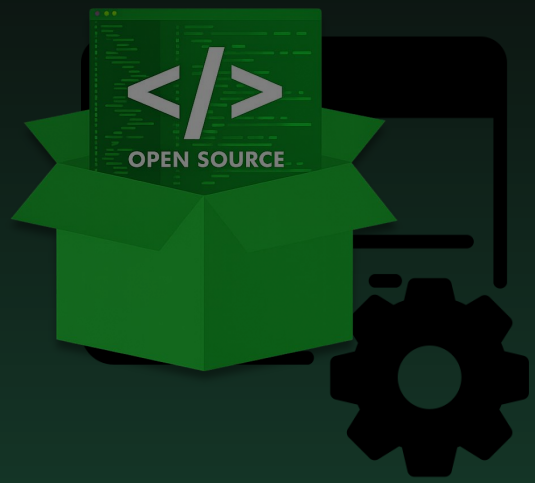


Build and manage databases

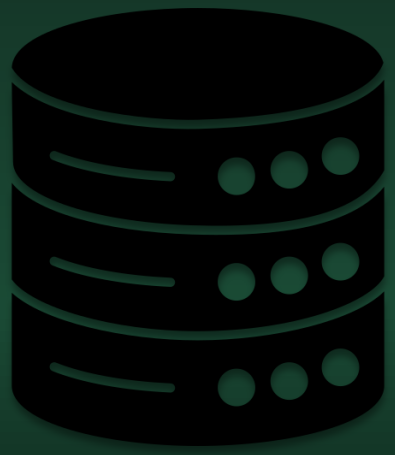
Handle user authentication

File storage

supabase



open-source backend platform



Build and manage databases

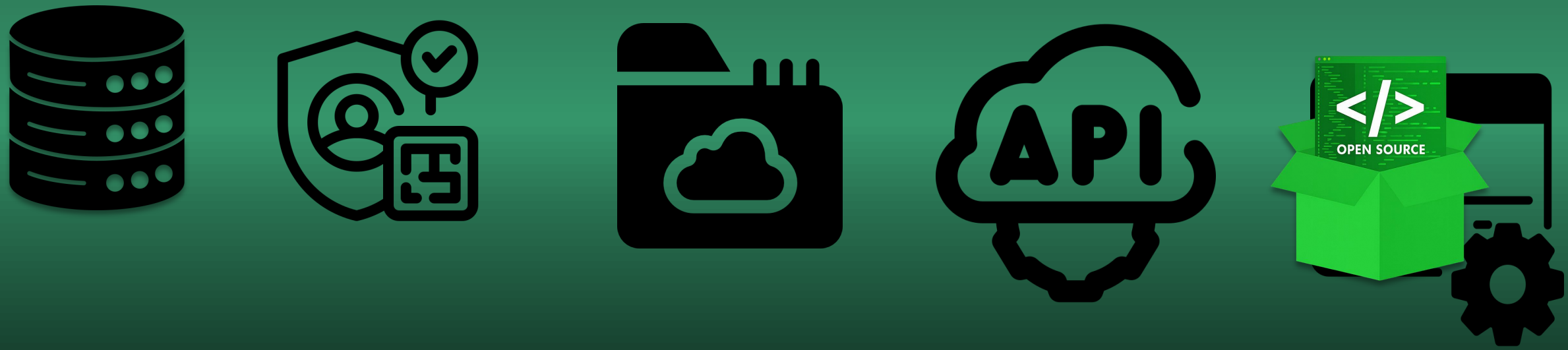
Handle user authentication

File storage

API requests

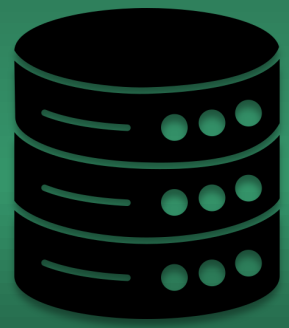
supabase

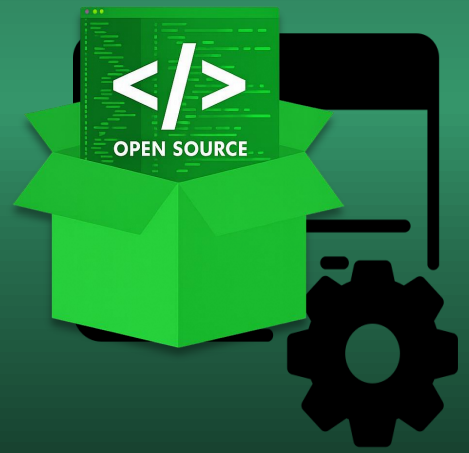
“**ready-to-use server**” for your projects,
built on top of a real database:
PostgreSQL

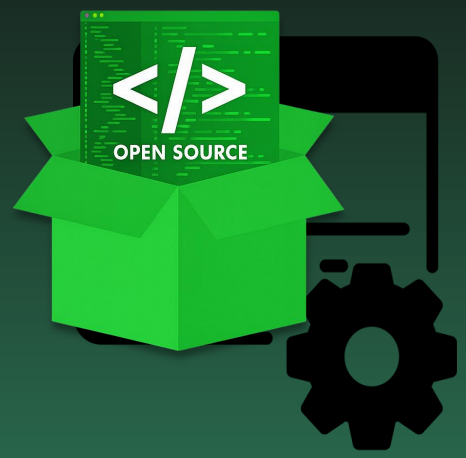


supabase

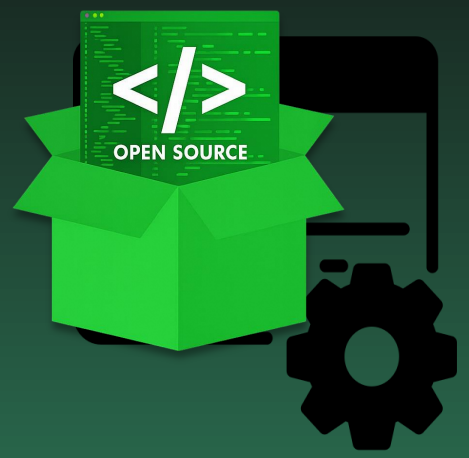
These function would allow the data scientist to focus on data analysis instead of server setup



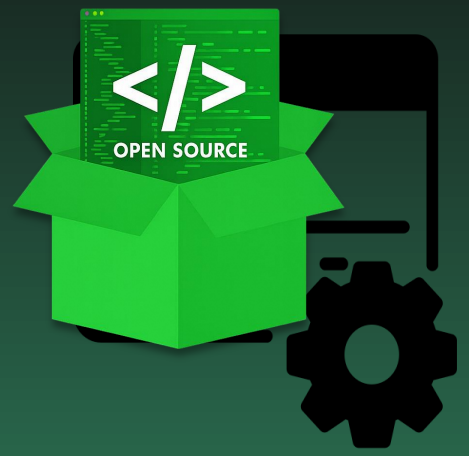




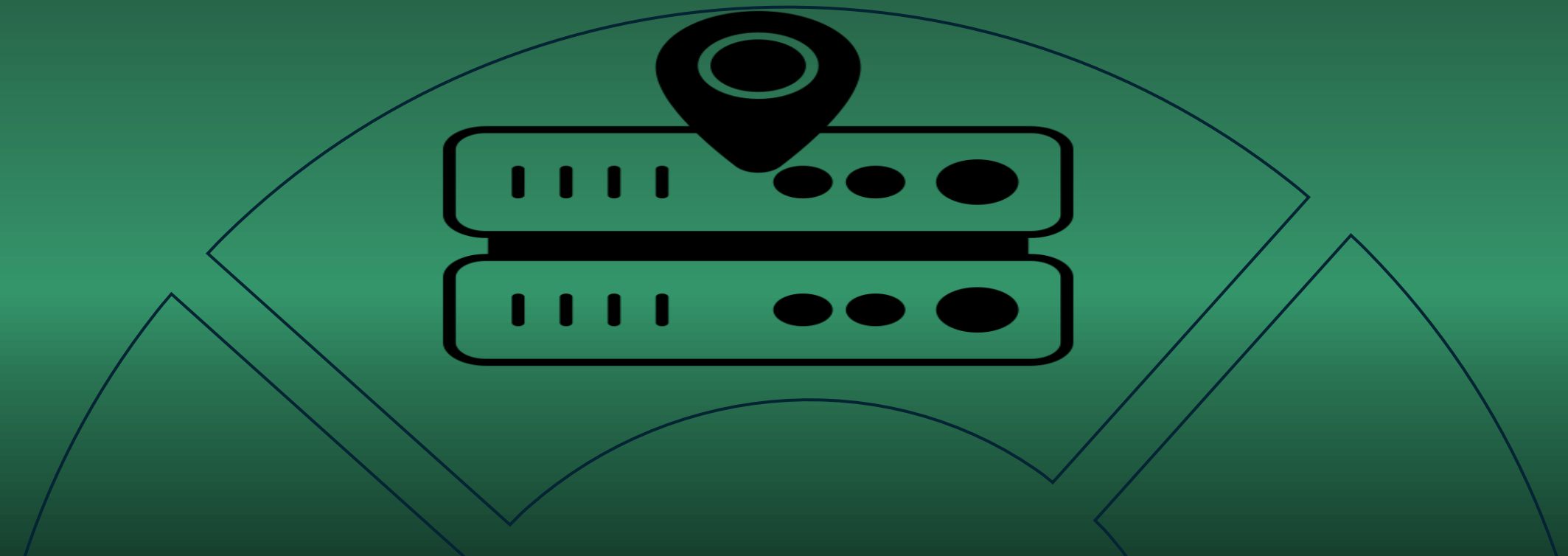
The open-source nature of the platform allows it to be hosted anywhere whether:

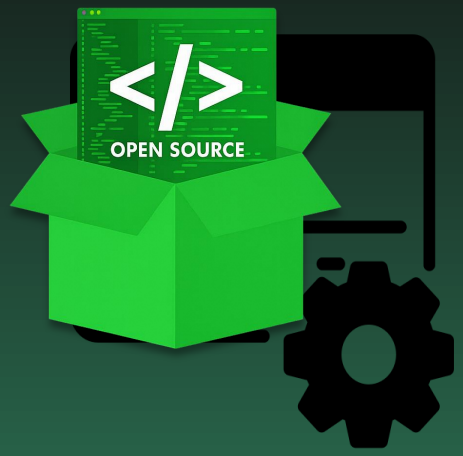


Any cloud provider that you feel comfortable with

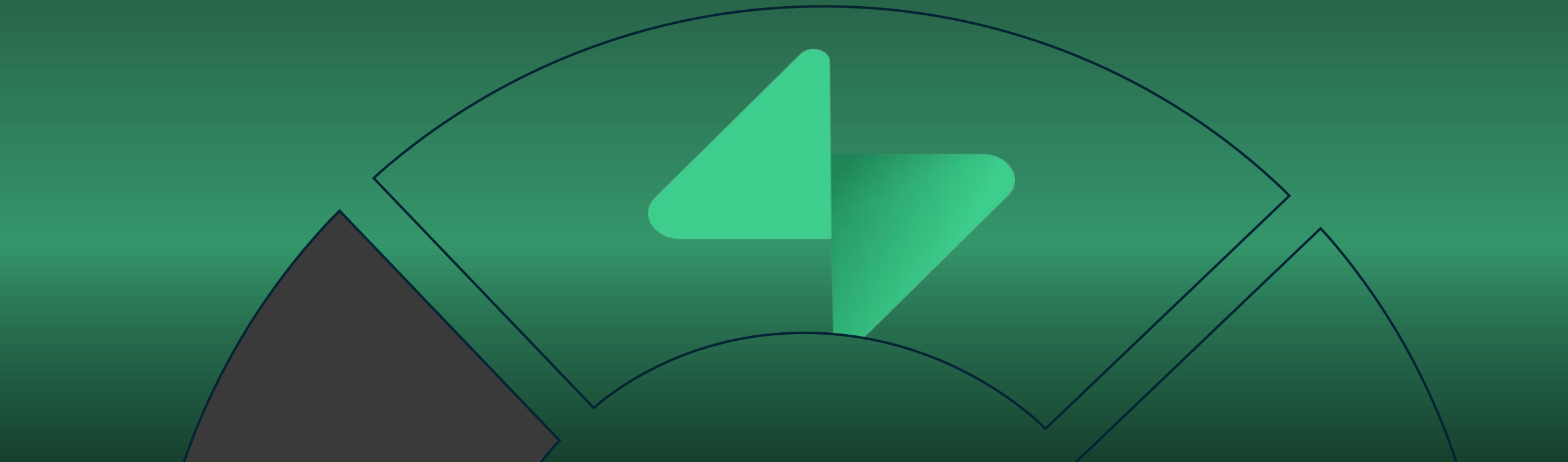


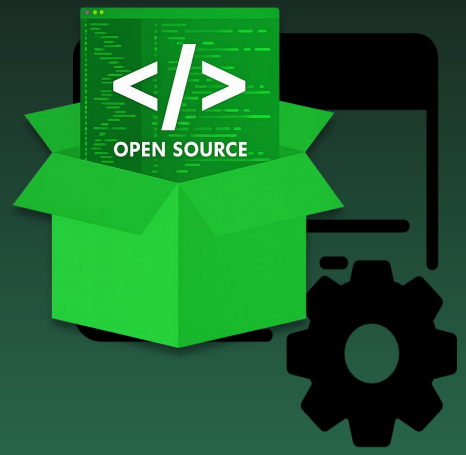
Your local infrastructure



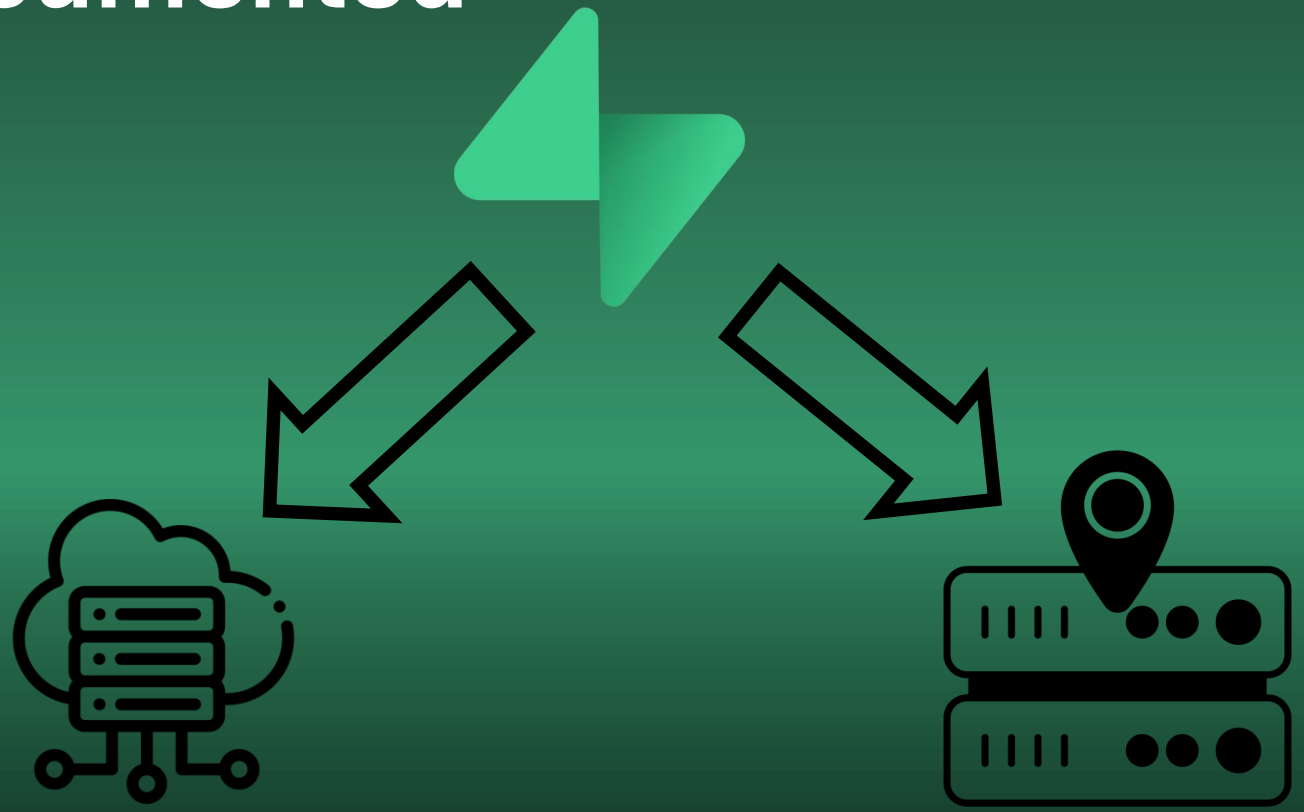


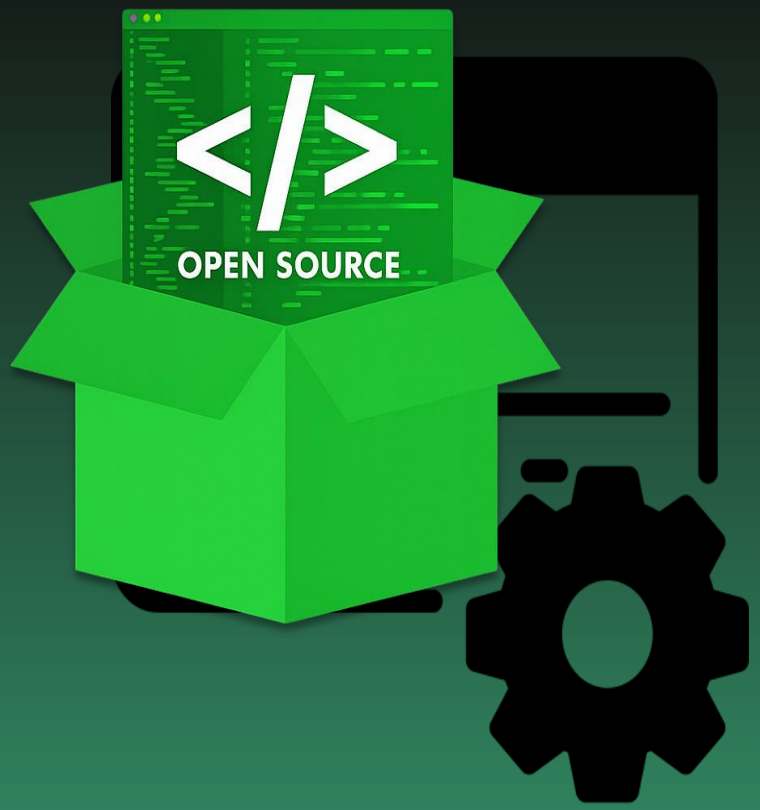
Simply use the official hosting. offering unlimited read, write API requests unlike other providers

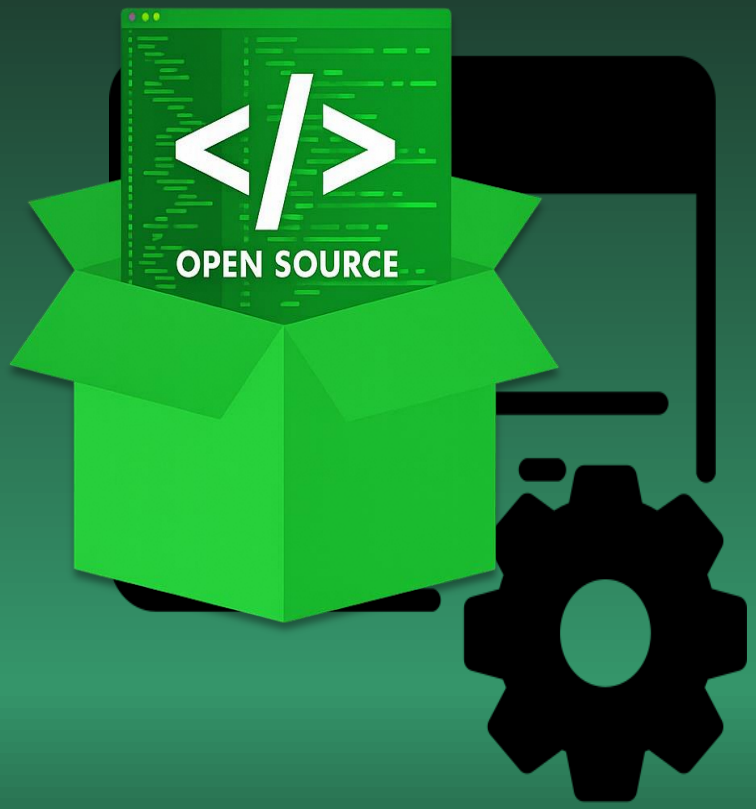




Or just use them all, the open-source nature makes the transition easy and well documented



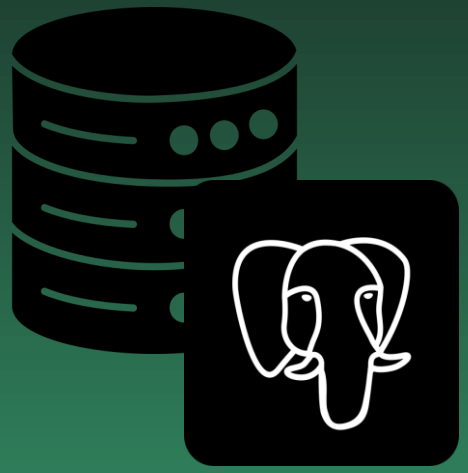

OPEN SOURCE

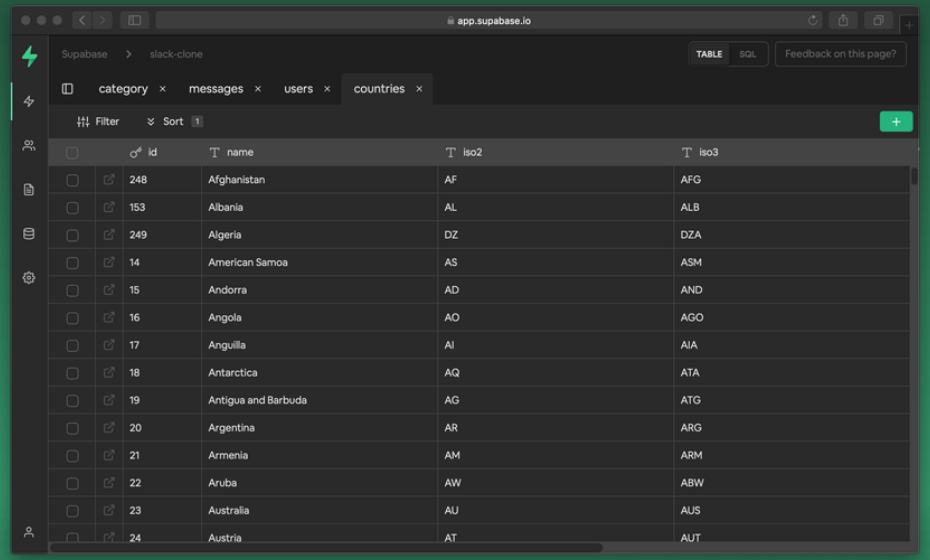

OPEN SOURCE

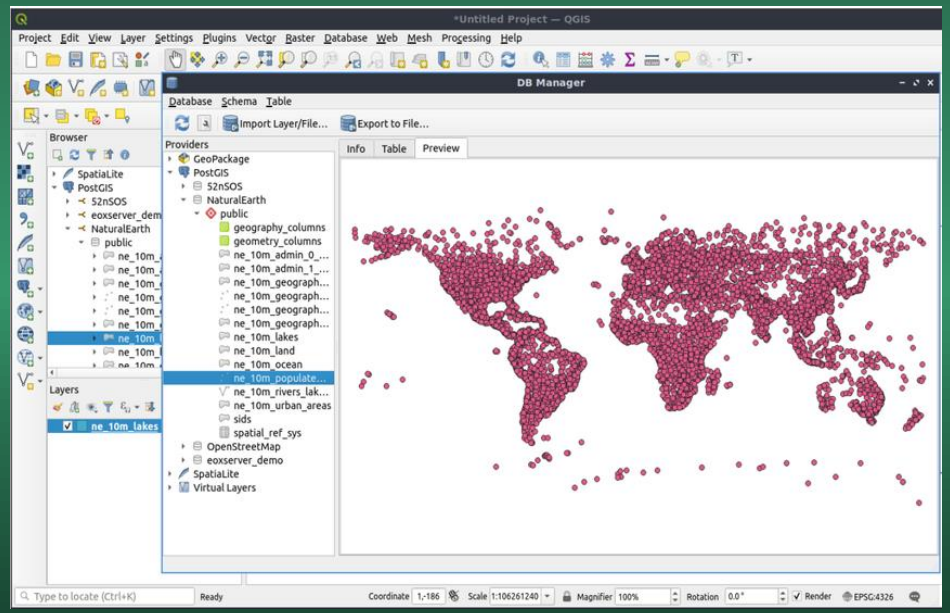
As a data scientist why should I care?

1. Database Management

1. Database Management







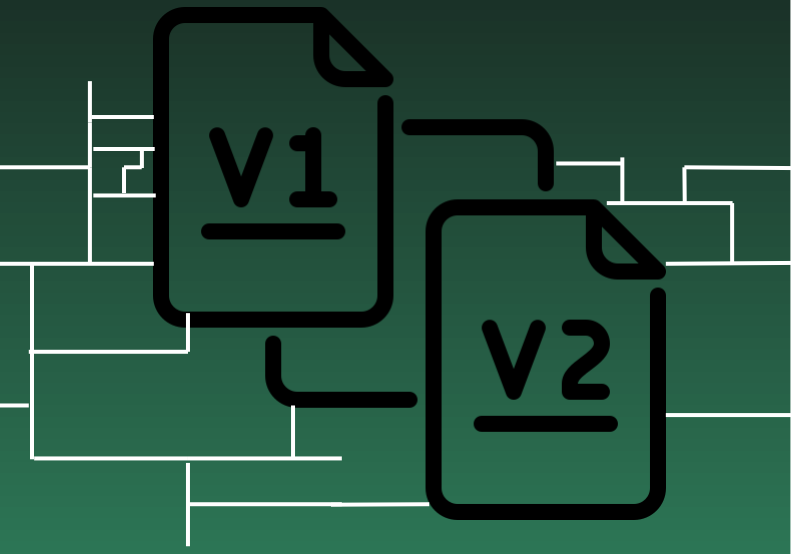
2. Multiple client Libraries support

```python
from supabase import create_client
import pandas as pd

url = "https://YOUR_PROJECT.supabase.co"
key = "YOUR_SUPABASE_KEY"
supabase = create_client(url, key)

# fetch data from table
data = supabase.table("sales").select("*").execute()
df = pd.DataFrame(data.data)
print(df.head())
```

2. Multiple client Libraries support

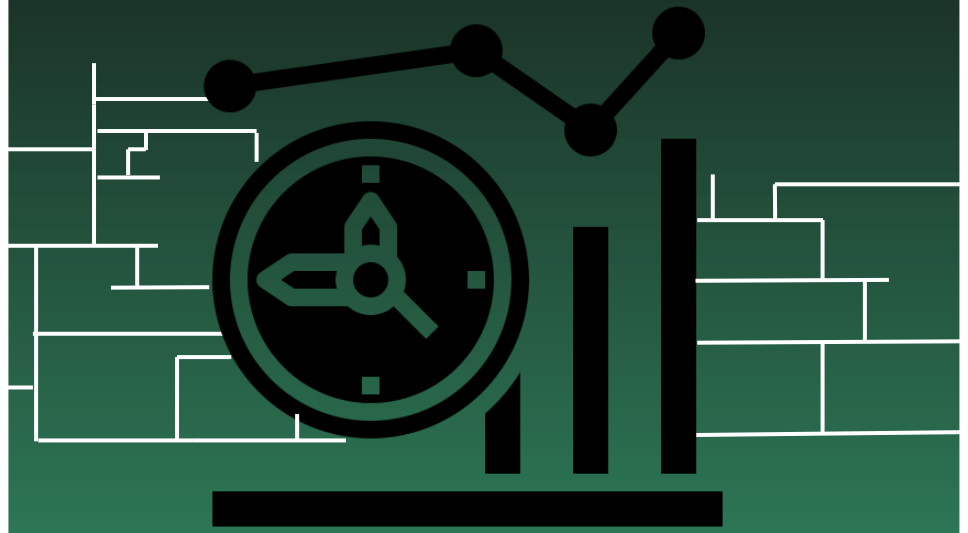

```python
from supabase import create_client
import pandas as pd

url = "https://YOUR_PROJECT.supabase.co"
key = "YOUR_SUPABASE_KEY"
supabase = create_client(url, key)

# fetch data from table
data = supabase.table("sales").select("*").execute()
df = pd.DataFrame(data.data)
print(df.head())
```

3.Data Versioning



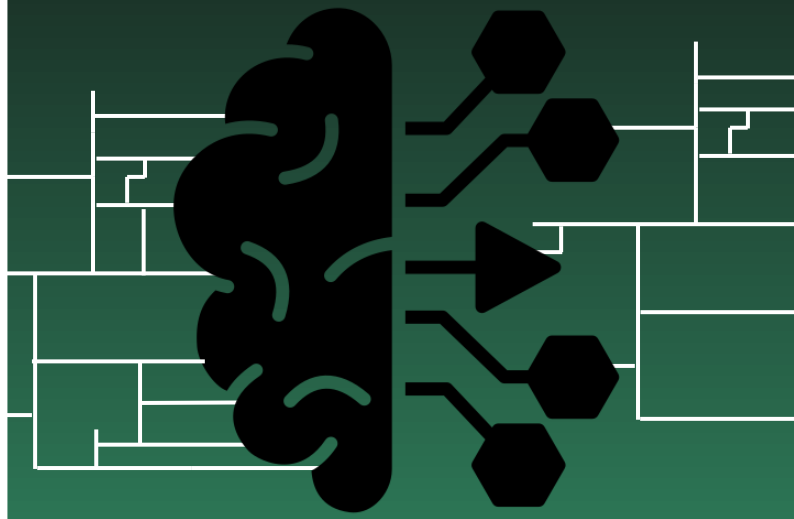
- Supabase supports **table snapshots, history tables, and audit logs**.
- You can keep multiple versions of datasets, which is useful **to visualize the changes in data**, and **for ML experiments**

4.Realtime Analytics



- **Supabase includes Realtime API**
- **that lets your app or data pipeline to automatically receive updates whenever data in the database changes.**

5. Integration With ML



- Supabase **can store embeddings or vector data** via **pgvector** extension for **AI/ML tasks.**
- ML model can fetch features for training or inference directly from the database

supabase