

# Lab : Installation Postgres

Pour l'installation de Postgre, je n'ai pas utilisé de VM. Étant sur un système Unix, je me suis dit que je ne devrai pas avoir de problème.

Pour ce faire, j'ai utilisé le gestionnaire de package de MacOS, homebrew pour télécharger et installer le SGBD.

```
gpin: Tue Dec 17 16:37:22 on ttys018
ve new mail.
rew install postgresql

wloading https://formulae.brew.sh/api/formula.json 100.0%
#####
wloading https://formulae.brew.sh/api/cask.json 100.0%
#####
wloading https://ghcr.io/v2/homebrew/core/postgresql/14/manifests/14.15
#####
tching dependencies for postgresql@14: icu4c@76, ca-certificates, openssl@3, krb5, lib and readline
wloading https://ghcr.io/v2/homebrew/core/icu4c/76/manifests/76.1.1
#####
tching icu4c@76 100.0%
wloading https://ghcr.io/v2/homebrew/core/icu4c/76/blobs/sha256:39c80ad78
#####
wloading https://ghcr.io/v2/homebrew/core/ca-certificates/manifests/2024-
#####
tching ca-certificates 100.0%
wloading https://ghcr.io/v2/homebrew/core/ca-certificates/blobs/sha256:7a
#####
wloading https://ghcr.io/v2/homebrew/core/openssl/3/manifests/3.4.0
#####
tching openssl@3 100.0%
wloading https://ghcr.io/v2/homebrew/core/openssl/3/blobs/sha256:786b2f22
#####
wloading https://ghcr.io/v2/homebrew/core/krb5/manifests/1.21.3
#####
tching krb5 100.0%
wloading https://ghcr.io/v2/homebrew/core/krb5/blobs/sha256:c7e8ec4458b77
#####
wloading https://ghcr.io/v2/homebrew/core/lib/manifests/1.10.0-1
#####
tching lib 100.0%
wloading https://ghcr.io/v2/homebrew/core/lib/blobs/sha256:698245d4c919
#####
wloading https://ghcr.io/v2/homebrew/core/readline/manifests/8.2.13
#####
tching readline 100.0%
wloading https://ghcr.io/v2/homebrew/core/readline/blobs/sha256:e4d4dfff0c
#####
tching postgresql@14 100.0%
wloading https://ghcr.io/v2/homebrew/core/postgresql/14/blobs/sha256:e0e
#####
stalling dependencies for postgresql@14: icu4c@76, ca-certificates, openssl@3, krb5, lib and readline
stalling postgresql@14 dependency: icu4c@76
wloading https://ghcr.io/v2/homebrew/core/icu4c/76/manifests/76.1.1
y downloaded: /Users/Mohameddjebali/Library/Caches/Homebrew/downloads/c105f898d3a6b8ff6253b47492e8652858048ee75fa52589cfc5b251b4d633---icu4c@76-76.1.1.bottle_manifest.json
uring icu4c@76-76.1.1.arm64_t8020_maverick.bottle.tar.gz
pt/homebrew/cellar/icu4c@76/76.1.1: 277 files, 81MB
stalling postgresql@14 dependency: ca-certificates
wloading https://ghcr.io/v2/homebrew/core/ca-certificates/manifests/2024-
y downloaded: /Users/Mohameddjebali/Library/Caches/Homebrew/downloads/c1b555434e8bc14721ffb41e1a71c1b853417578c631fa1b69e8738f8cafc76c---ca-certificates-2024-11-26.bottle_manifest.json
uring ca-certificates-2024-11-26.all.bottle.tar.gz
generating CA certificate bundle from keychain, this may take a while...
pt/homebrew/cellar/ca-certificates/2024-11-26: 4 files, 239.4KB
stalling postgresql@14 dependency: openssl@3
wloading https://ghcr.io/v2/homebrew/core/openssl/3/manifests/3.4.0
y downloaded: /Users/Mohameddjebali/Library/Caches/Homebrew/downloads/a8a169d38329b14f967d49708ea77ee6dc5444175093fc7b26826b357e81736---openssl@3-3.4.0.bottle_manifest.json
uring openssl@3-3.4.0.arm64_t8020_maverick.bottle.tar.gz
pt/homebrew/cellar/openssl@3/3.4.0: 7,236 files, 33.4MB
stalling postgresql@14 dependency: krb5
wloading https://ghcr.io/v2/homebrew/core/krb5/manifests/1.21.3
```

Installation

Puis, j'ai vérifié que le service était bien lancé.

```
psql                                psql -U user1 -d webappdb ...

Last login: Wed Dec 18 11:49:02 on ttys001
You have new mail.
+ ~ brew services list

Name      Status User      File
php       started mohameddjebali ~/Library/LaunchAgents/homebrew.mxcl.php.plist
postgresql@14 started mohameddjebali ~/Library/LaunchAgents/homebrew.mxcl.postgresql@14.plist
symfony-cli none
+ ~
```

Service lancé

Suite à ça, j'ai lancé les différentes commandes pour créer un utilisateur et une BDD. Ici, nous voyons user1 associé à la DB webappdb.

```
webappdb=> \l webappdb
```

List of databases					
Name	Owner	Encoding	Collate	Ctype	Access privileges
webappdb	user1	UTF8	C	C	=Tc/user1 + user1=CTc/user1

(1 row)

# Lab : Visualization des données avec Grafana

Pour la création de oildb, j'ai suivi les mêmes étapes que sur le lab précédent.

Sur l'image ci-dessous, on voit sur la première ligne que je crée la DB et que je l'affecte à mon user1.

Ensuite, par l'intermédiaire du user1, je lance la query me permettant de créer la table.

```
[postgres=# CREATE DATABASE oildb OWNER user1;
CREATE DATABASE
postgres=# psql -U user1 -d oildb
postgres=#
\q
→ ~ psql -U user1 -d oildb

psql (14.15 (Homebrew))
Type "help" for help.

oildb=> CREATE TABLE t_oil(
    region TEXT,
    country TEXT,
    year INT,
    production INT,
    consumption INT
);
CREATE TABLE
oildb=> grant all privileges on database "oildb" to "user1";
ERROR:  database ""oildb"" does not exist
oildb=> grant all privileges on database oildb to user1;
GRANT
oildb=> █
```

Dans le screenshot ci-dessous, on voit que dans ma première tentative de copie du fichier, on me retourne une erreur. Mais on me donne un petit indice pour pouvoir copier simplement.

💡 HINT: Anyone can COPY to stdout or from stdin. psql's \copy command also works for anyone.

J'utilise donc la commande \copy.

Suite à cela, j'ai eu une autre erreur. En effet, je n'ai pas choisi le bon delimiter et on me signale subtilement que c'est un « ; » dont j'ai besoin.

```

oildb=> COPY t_oil(region, country, year, production, consumption)
[FROM 'Users/mohameddjebali/Downloads/oil-data.csv'
DELIMITER ',' CSV HEADER;
ERROR: must be superuser or a member of the pg_read_server_files role to COPY from a file
HINT: Anyone can COPY to stdout or from stdin. psql's \copy command also works for anyone.
oildb=> \COPY t_oil(region, country, year, production, consumption)
FROM '/Users/mohameddjebali/Downloads/oil-data.csv'
DELIMITER ',' CSV HEADER;
ERROR: missing data for column "country"
CONTEXT: COPY t_oil, line 2: "North America;USA;1966;9579;12100"
[oildb=> SELECT * FROM t_oil;
 region | country | year | production | consumption
-----+-----+-----+-----+-----
(0 rows)

oildb=> \COPY t_oil(region, country, year, production, consumption)
FROM '/Users/mohameddjebali/Downloads/oil-data.csv'
[DELIMITER ';' CSV HEADER;
COPY 164
[oildb=> SELECT * FROM t_oil;
oildb=> █

```

Suite à cela, afin de vérifier que mes données ont été correctement importées, je lance une query pour consulter la table.

region	country	year	production	consumption
North America	USA	1966	9579	12100
North America	USA	1967	10219	12567
North America	USA	1968	10600	13405
North America	USA	1969	10828	14153
North America	USA	1970	11297	14710
North America	USA	1971	11156	15223
North America	USA	1972	11185	16381
North America	USA	1973	10946	17318
North America	USA	1974	10461	16631
North America	USA	1975	10008	16334
North America	USA	1976	9736	17461
North America	USA	1977	9863	18443
North America	USA	1978	10274	18756
North America	USA	1979	10136	18438
North America	USA	1980	10170	17062
North America	USA	1981	10181	16060
North America	USA	1982	10199	15295
North America	USA	1983	10247	15235
North America	USA	1984	10509	15725
North America	USA	1985	10580	15726
North America	USA	1986	10231	16281

Pour l'installation de Grafana, je me suis encore servi de mon gestionnaire de package sur MacOS, Homebrew.

```
→ ~ brew update

==> Updating Homebrew...
Already up-to-date.
→ ~ brew install grafana

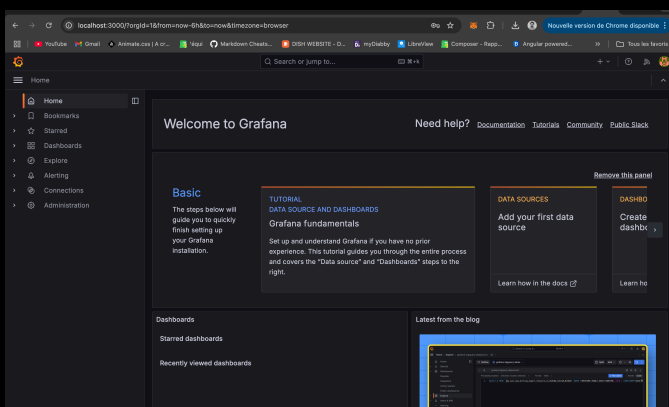
==> Downloading https://ghcr.io/v2/homebrew/core/grafana/manifests/11.4.0
#####
==> Fetching grafana
==> Downloading https://ghcr.io/v2/homebrew/core/grafana/blobs/sha256:328dfb28cbd6f001c1bd02d6cafd64783237450
#####
==> Pouring grafana--11.4.0.arm64_sonoma.bottle.tar.gz
==> Caveats
To start grafana now and restart at login:
  brew services start grafana
Or, if you don't want/need a background service you can just run:
  /opt/homebrew/opt/grafana/bin/grafana server --config /opt/homebrew/etc/grafana/grafana.ini --homepath /opt/homebrew/opt/grafana --log/grafana cfg:default.paths.data=/opt/homebrew/var/lib/grafana cfg:default.paths.plugins=/opt/homebrew/opt/grafana/var/lib/grafana/plugins
==> Summary
📦 /opt/homebrew/Cellar/grafana/11.4.0: 8,062 files, 427.2MB
==> Running `brew cleanup grafana`...
Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.
Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).
→ ~
```

J'ai vérifié si le service était bien lancé :

```
Error: Invalid usage: unknown subcommand: `status`
[+] ~ brew services start grafana

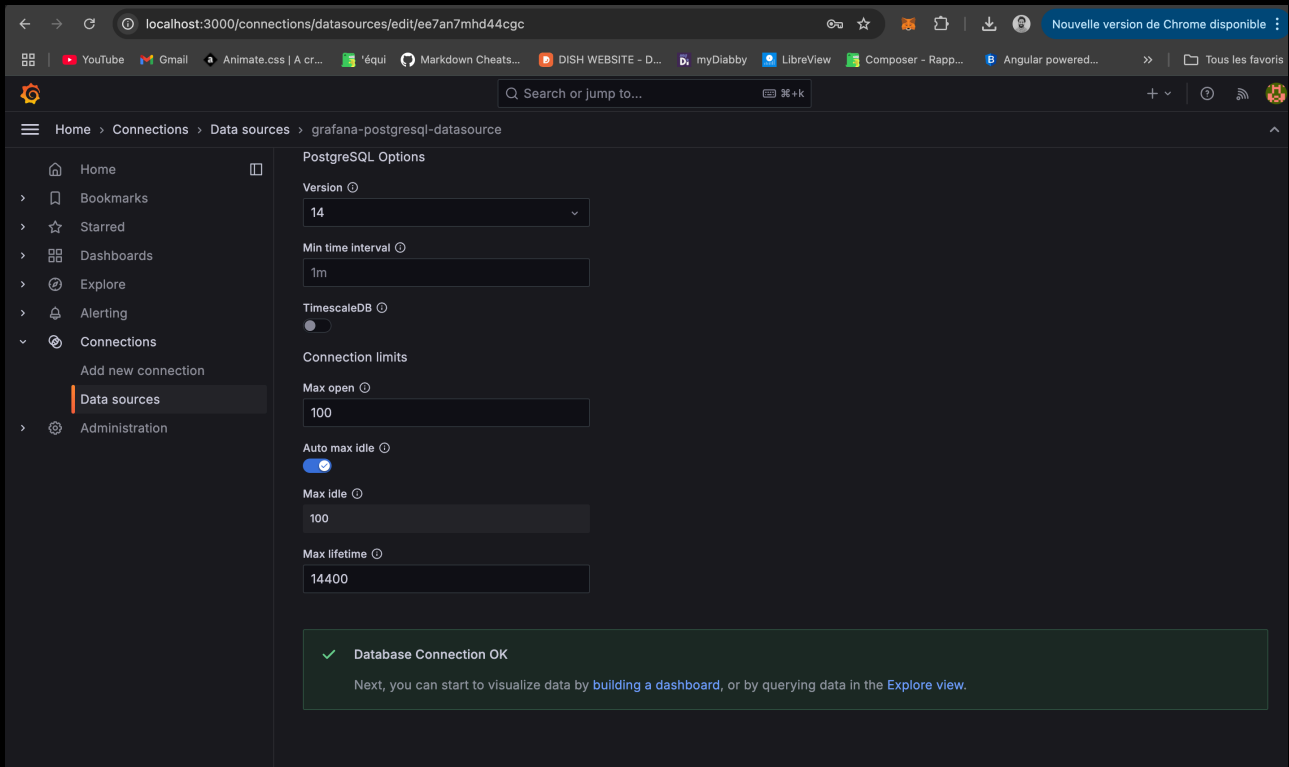
Service `grafana` already started, use `brew services restart grafana` to restart.
→ ~
```

Enfin, j'ai lancé Grafana qui écoute par défaut sur le port 3000 :

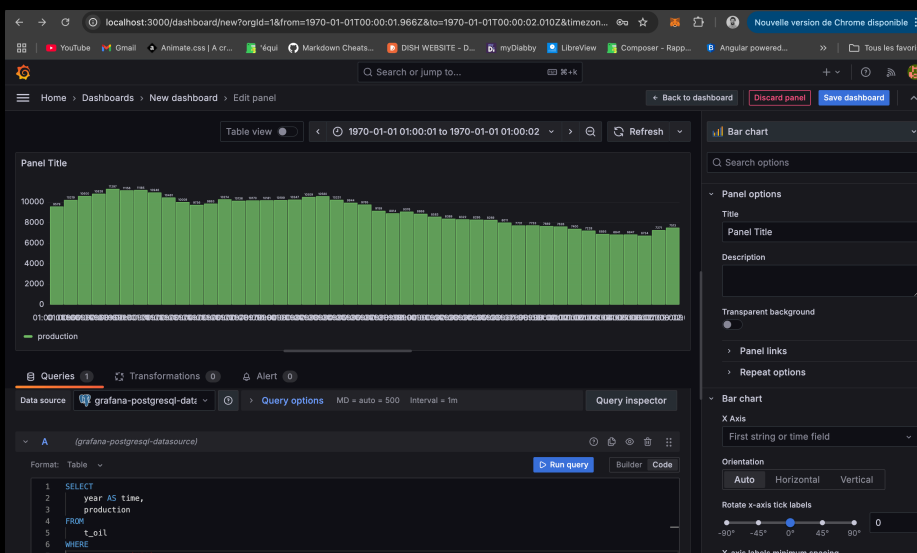




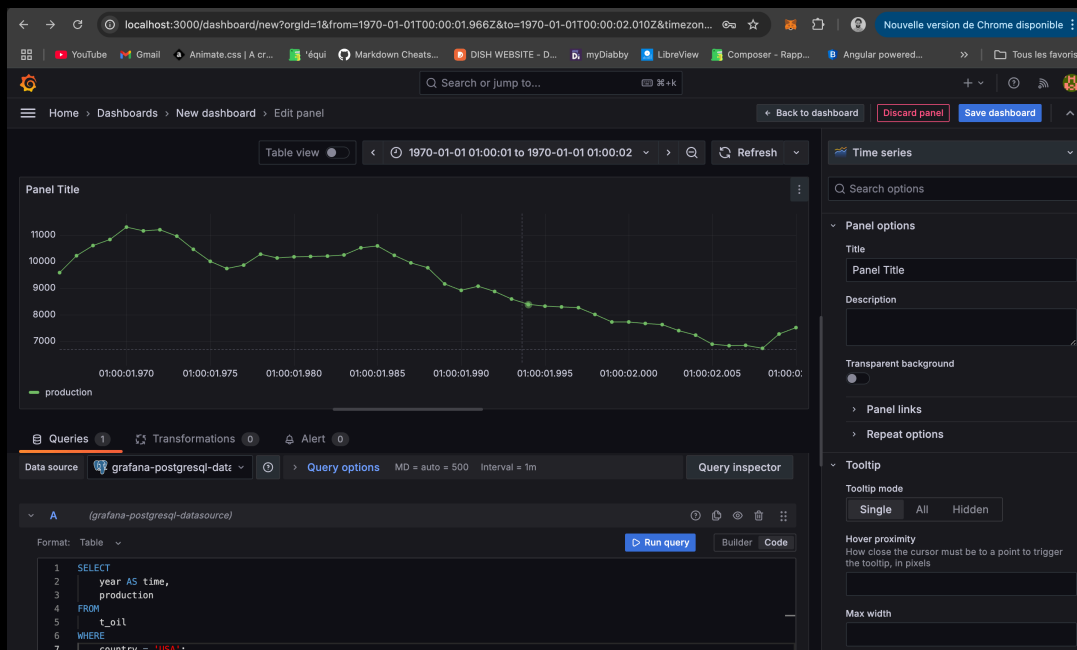
Suite à cela j'ai configuré sur l'interface de Grafana, la connection à la base de données PostgreSQL.



Sur l'image ci-dessous, j'ai choisi le bar chart comme type de visualisation :



Pour ma part, le meilleure graphique pour la visualisation de ces données serait le Time Series. Je le trouve beaucoup plus clair pour visualiser les tendances et variation sur une période étendue.



Enfin, pour avoir un graphe qui permet de visualiser la production et la consommation de chaque pays, j'ai utilisé le Time Series et j'ai modifié ma requête SQL pour intégrer les pays que je souhaitais voir.

