Инструкция по установке APACHE SUPERSET в DOCKER

ver 9 (26.12.2023)

Оглавление

[**1.** **Установка  Docker Desktop:** 2](#_Toc154481714)

[**2.** **Установка  SUPERSET** 4](#_Toc154481715)

[**3.** **Установка Postgres (по желанию)** 11](#_Toc154481716)

[**4.** **ПРИЛОЖЕНИЕ** 21](#_Toc154481717)

[**5.** **Установка superset** 22](#_Toc154481718)

[**6.** **IMPALA** 29](#_Toc154481719)

[**7.** **VAR 3** 30](#_Toc154481720)

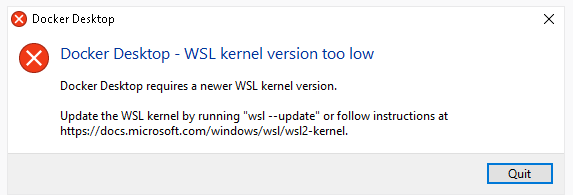
# **Установка  Docker Desktop:**

<https://www.docker.com/products/docker-desktop/>

**Для Windows:**

https://docs.docker.com/desktop/install/windows-install/

Обновить WSL если потребуется:

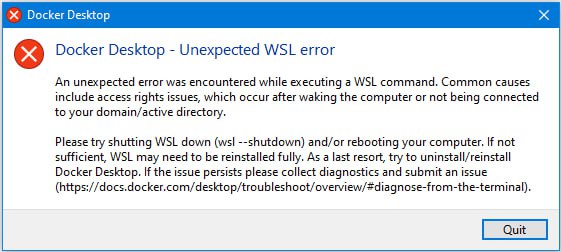


Включить виртуализацию в БИОС, если выдает подобное предупреждение:



При установке Докера в windows отказаться от WSL по умолчанию (снять/поставить галку)

Если такое сообщение:

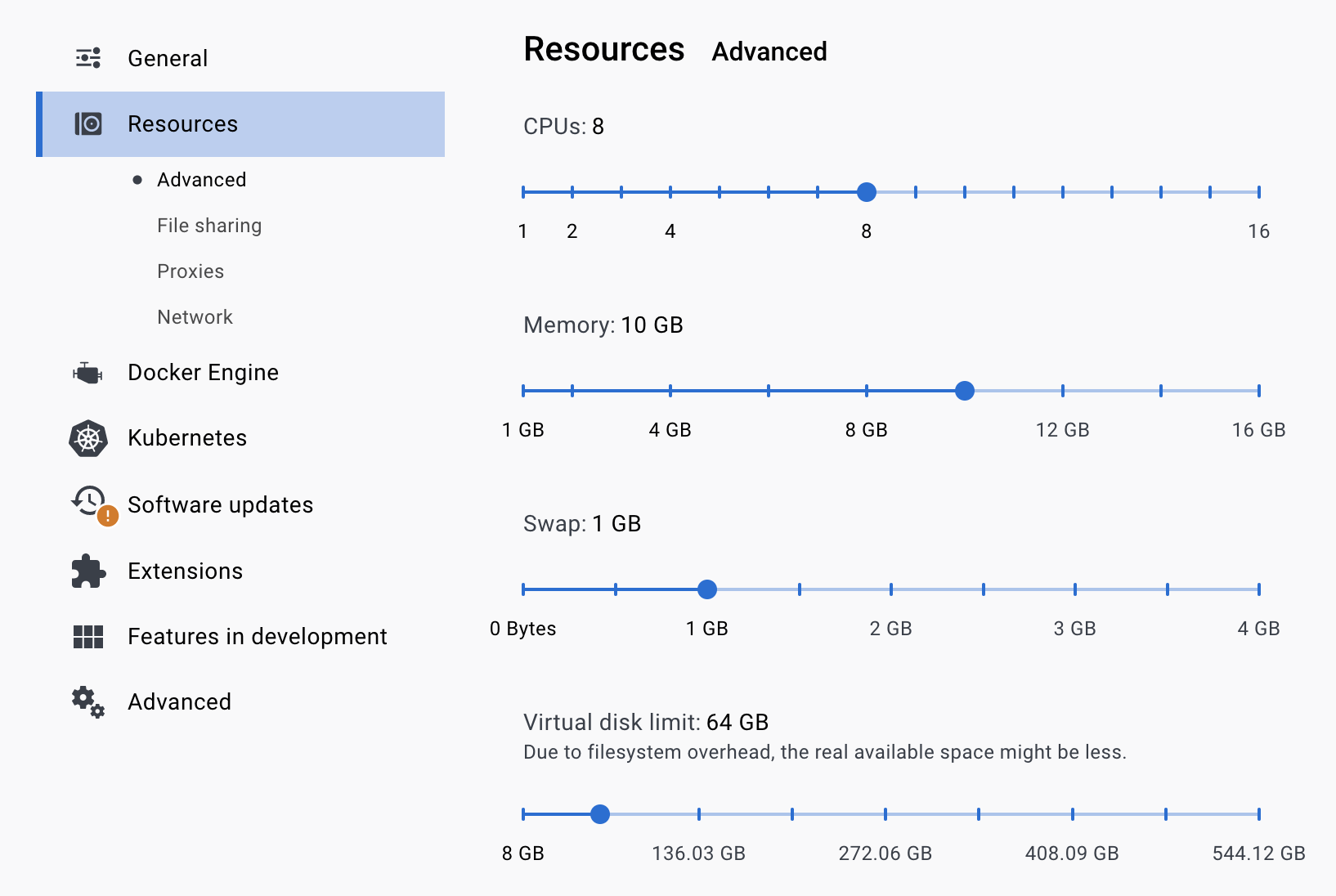


То выполнить

wsl —shutdown

**Добавить докеру больше ресурсов памяти и ядер.**

Зайти в настройки докера и увеличить:



После работы с docker его желательно выключать, так как он использует дополнительные ресурсы.

**Запустить Docker Desktop**

# **Установка  SUPERSET**

Убедиться, что никакие сервисы не занимают порт 8080. (например jupyter notebook)

Перейти по ссылке и ничего не должно открыться:

[http://localhost:8080](http://localhost:8080/)

#### Открыть ТЕРМИНАЛ:

#### Для windows (Win+R, затем ввести cmd затем Enter)

#### 

#### Если в windows не выполняются команды в терминале, то использовать powershell <https://dzen.ru/a/XzVQXM0dchqol_jh>

#### Так же часть проблем можно решить запуском коммандера от имени администратора:

#### Поиск-командная строка-правая кнопка мыши-запуск от имени администратора

**Для пользователей Mac M1.**

Перед первой командой необходимо ввести в терминале команду:  
**export DOCKER\_DEFAULT\_PLATFORM=linux/amd64**

Далее идти по инструкции

Через Safari логин и пароль не вводились, заходит только через Chrome.

Выполнить команду:

docker pull apache/superset

Будет скачиваться образ в течении нескольких минут

Выполнить команду:

openssl rand -base64 42

Если у вас Windows, то можете не выполнять openssl, можно сгенерировать вручную любой ключ или оставить этот, это все равно локальная установка.

Например такой:

ZLfWZYSYmB1LBDf16ozTBgDoAHVKWFgr96I5lAUElNPFeMDiU3USrQEn

Этот ключ вставить в команду ниже:

Выполнить команду:

docker run -d -p 8080:8088 -e "SUPERSET\_SECRET\_KEY= ZLfWZYSYmB1LBDf16ozTBgDoAHVKWFgr96I5lAUElNPFeMDiU3USrQEn" --name superset apache/superset

Выполнить команду:

docker exec -it superset superset fab create-admin --username **admin** --firstname Superset --lastname Admin --email admin@admin.com --password **postgres123**

Убедиться, что команда выполнилась корректно.

Выполнить команду:

docker exec -it superset superset db upgrade

Выполнить команду: (скачиваются примеры датасетов и визуализаций)

docker exec -it superset superset load\_examples

Выполнить команду:

docker exec -it superset superset init

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Если выдает ошибку и пишет:

Error response from daemon: Conflict. The container name "/superset" is already in use by container "137493a\*\*\*\*\*\*\*\*3a77a". You have to remove (or rename) that container to be able to reuse that name.

docker run -d -p 8080:8088 -e "SUPERSET\_SECRET\_KEY= ZLfWZYSYmB1LBDf16ozTBgDoAHVKWFgr96I5lAUElNPFeMDiU3USrQEn" --name superset5 apache/superset

docker exec -it superset5 superset fab create-admin --username admin --firstname Superset --lastname Admin --email [admin@admin.com](mailto:admin@admin.com) --password postgres123

docker exec -it superset5 superset db upgrade

docker exec -it superset5 superset load\_examples

docker exec -it superset5 superset init

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Перейти по адресу: (обратите внимание - другие сервисы, как jupyter notebook не должны занимать этот порт)

http://localhost:8080

Ввести:

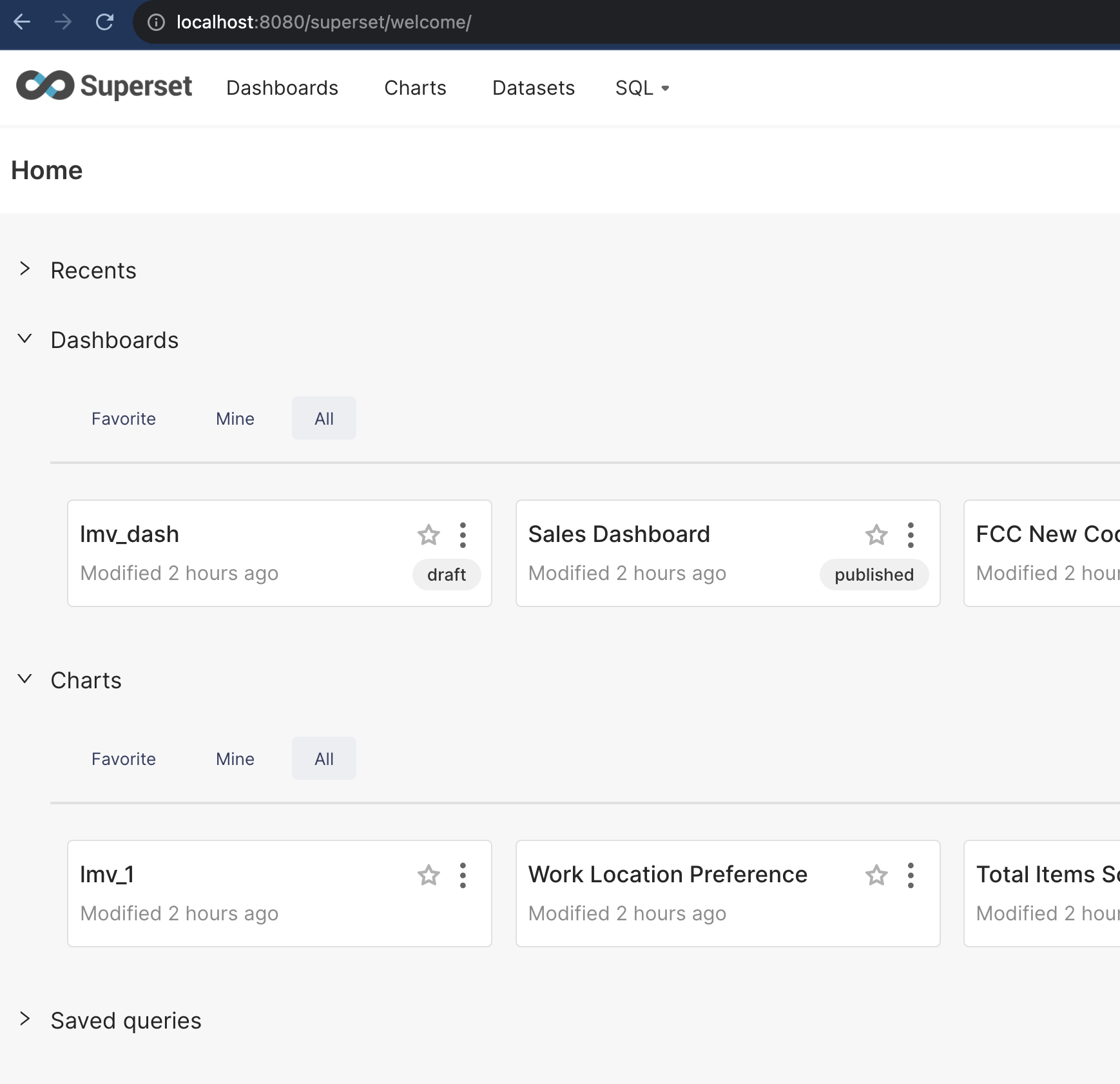
**ЛОГИН** admin

**ПАРОЛЬ** postgres123

Если логин пароль не подошел, значит команда ниже выполнилась с ошибкой, запустить заново:

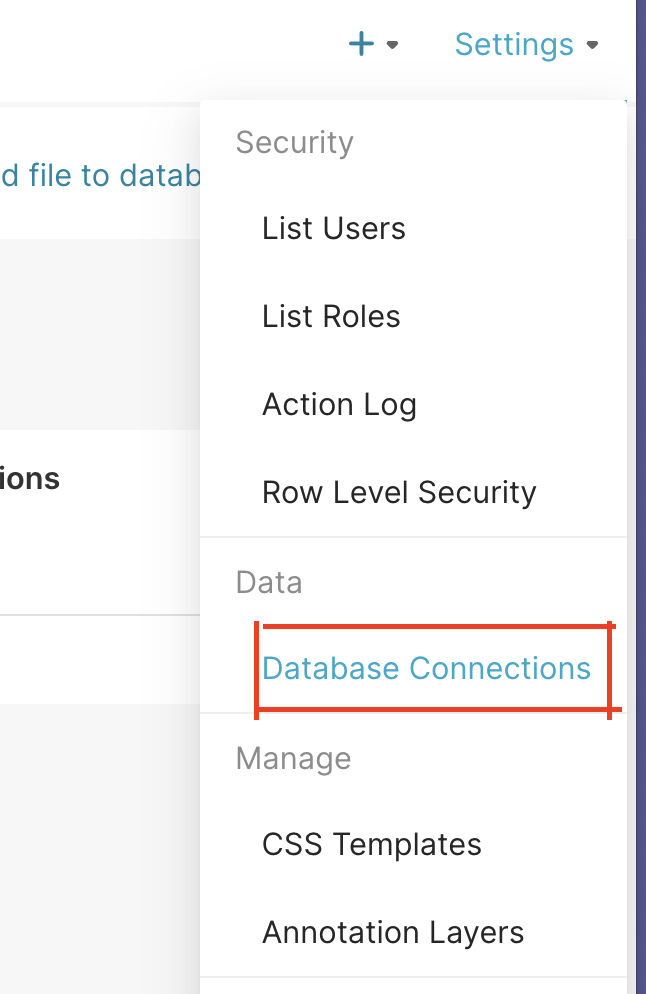
docker exec -it superset superset fab create-admin --username **admin** --firstname Superset --lastname Admin --email admin@admin.com --password **postgres123**

docker exec -it superset superset init

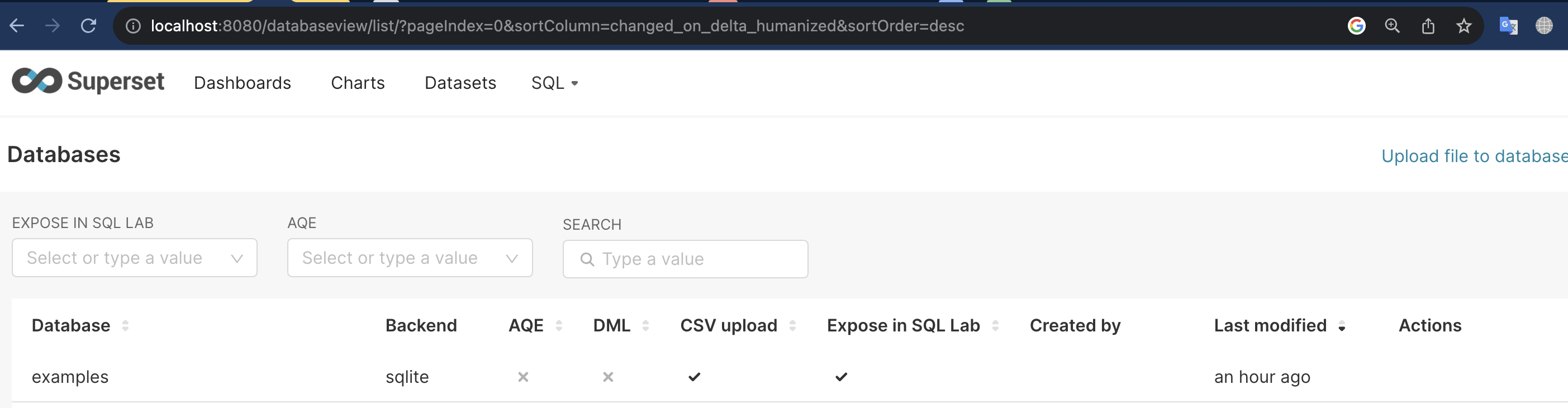


Можно перейти в:

Settings - DataBase Connections

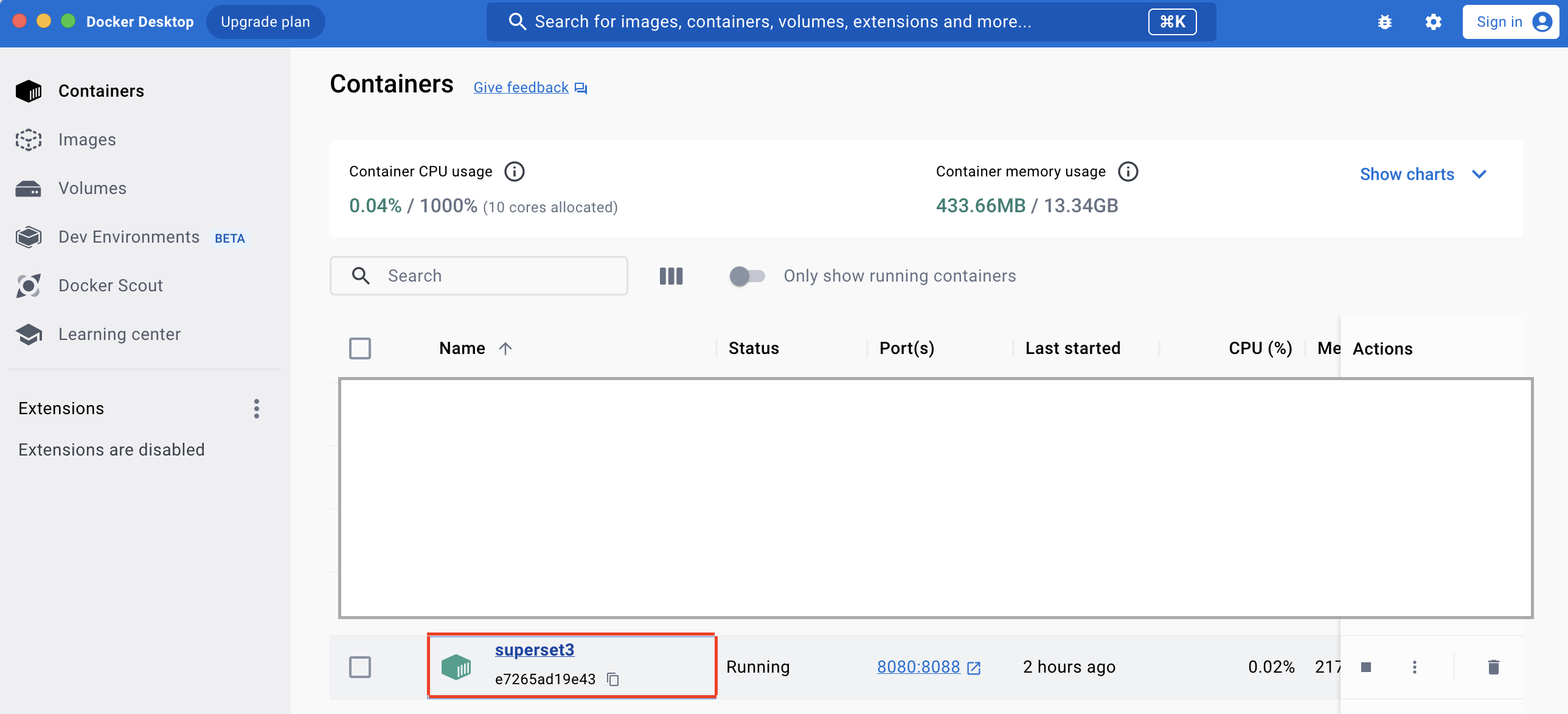


Будет доступна локальная база sqlite



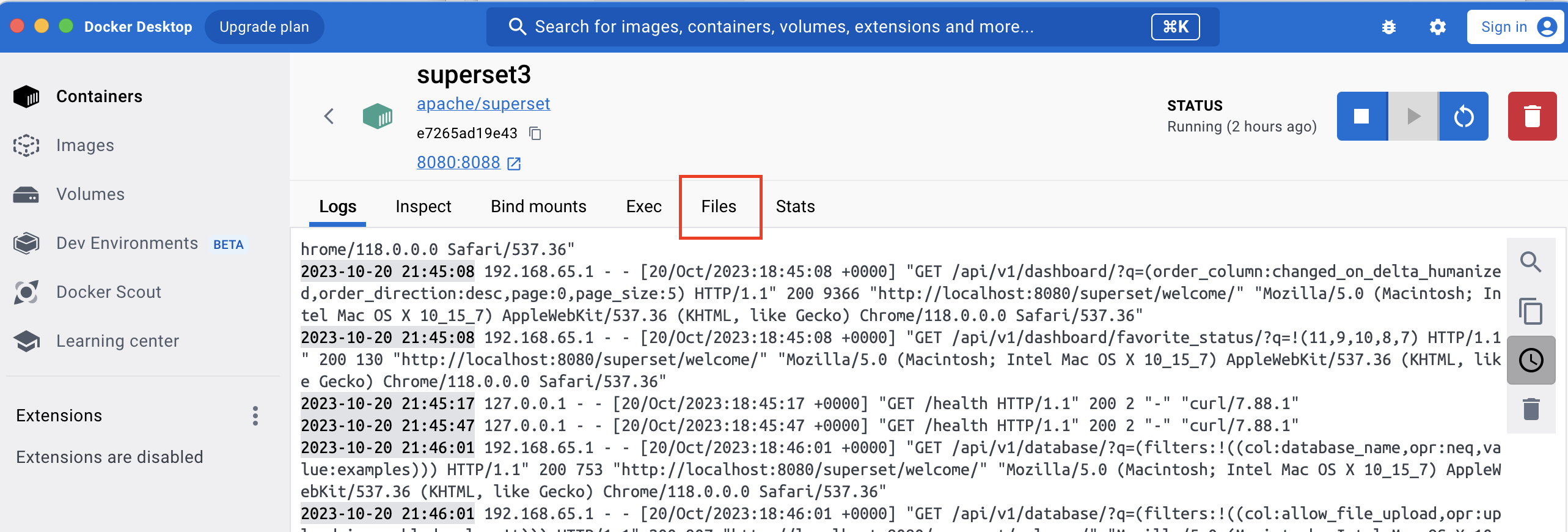
Чтобы в нее можно было загрузить csv файлы нужно:

Найти в Docker Desktop контейнер Superset



Кликнуть в его название, тогда в него можно будет зайти:

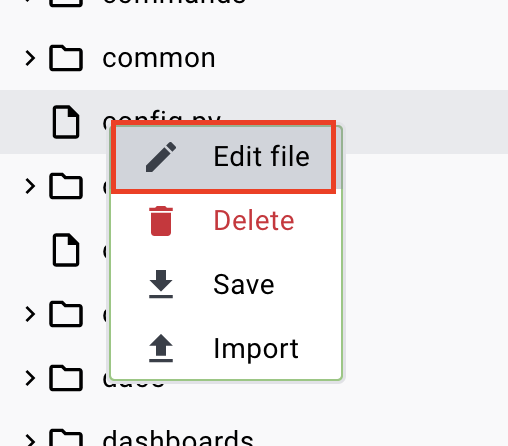
и выбрать меню Files:



Найти файл в папке:

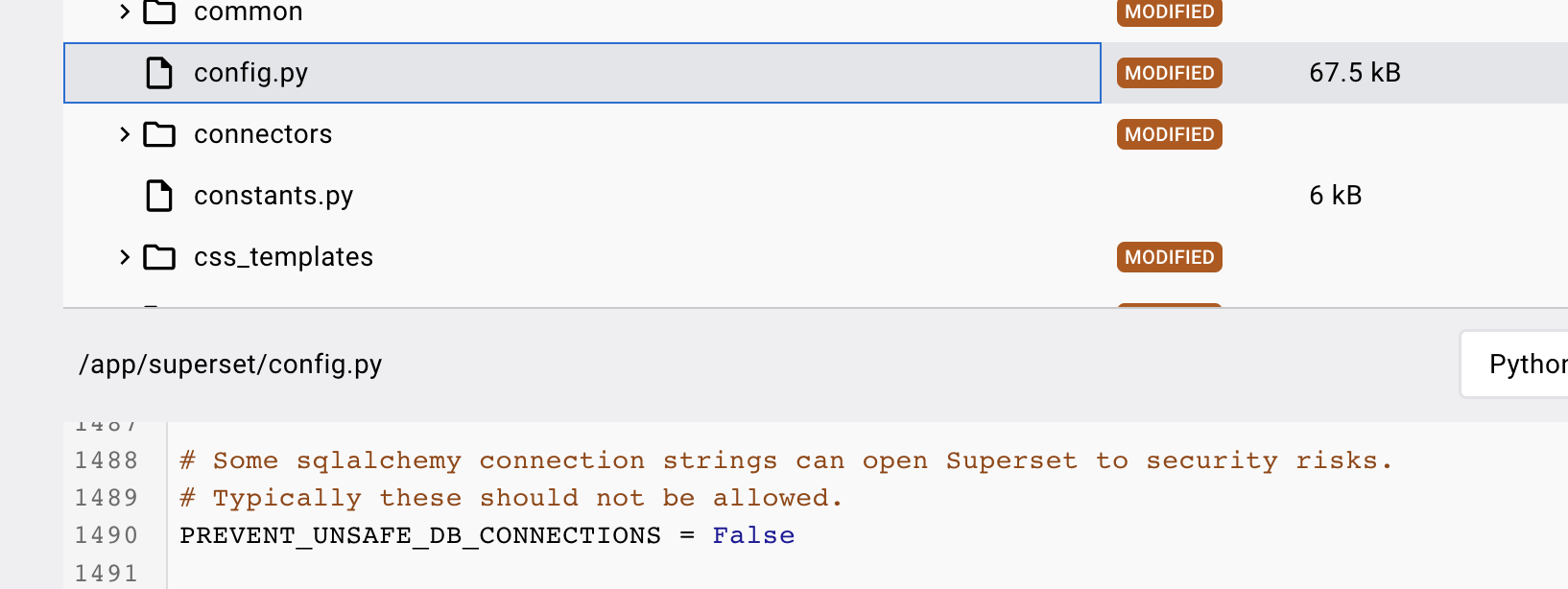
/app/superset/config.py

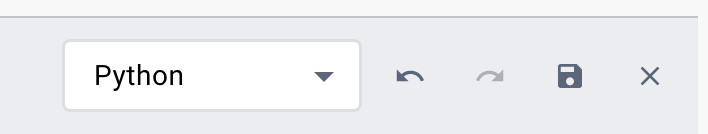
И ПКМ -> Edit file



Найти строку PREVENT\_UNSAFE\_DB\_CONNECTIONS = TRUE

и заменить ее на False



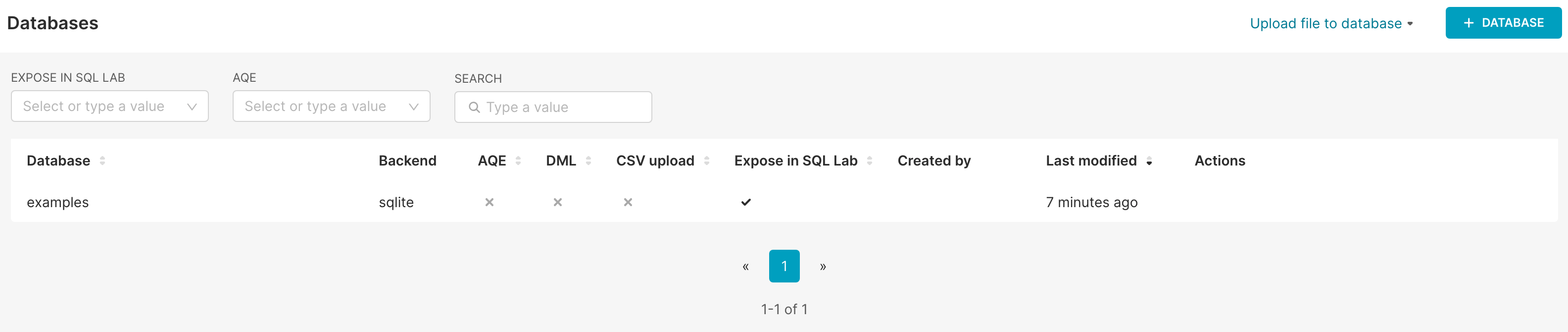


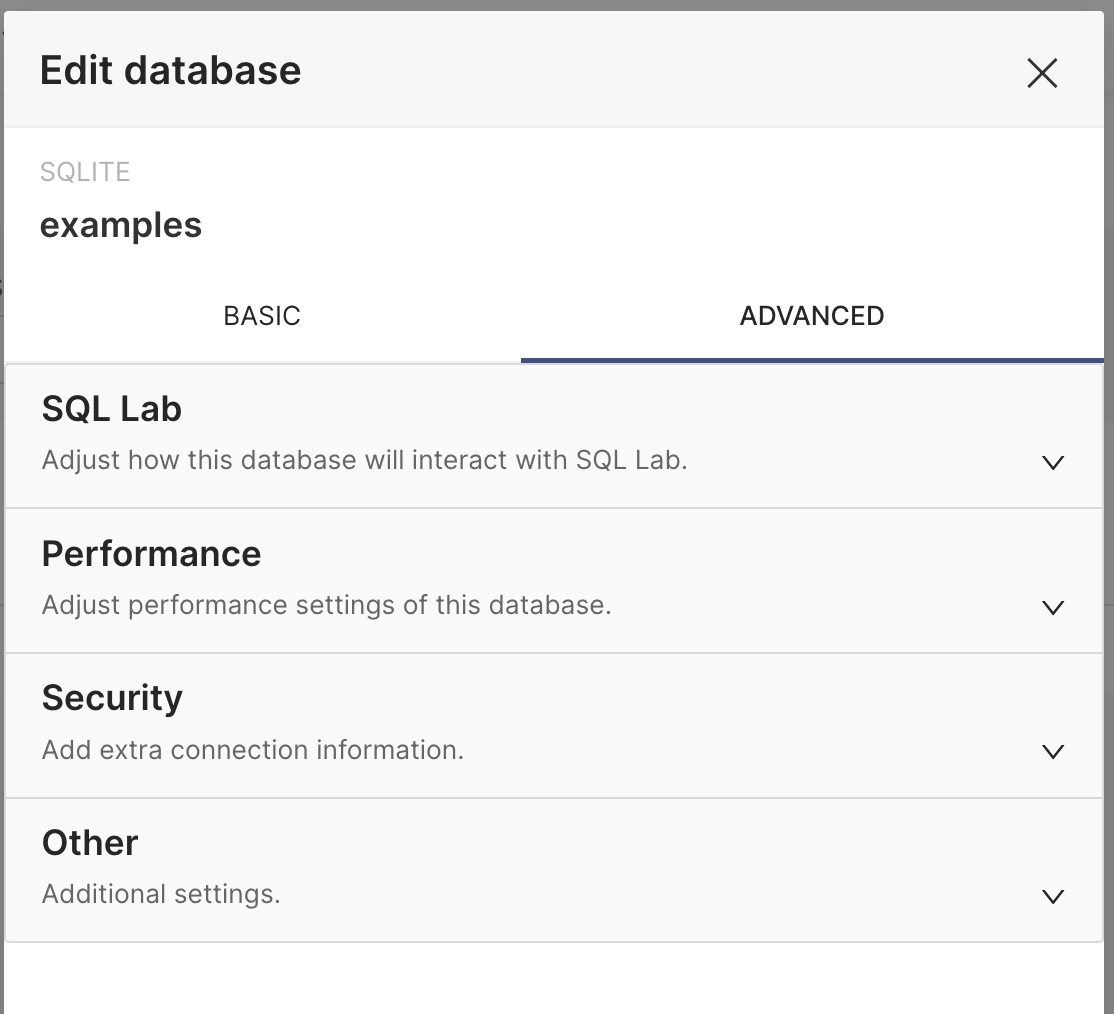
Save changes

Перезагрузить  суперсет (в консоли Docker)

Перейти в Settings - DataBase Connections

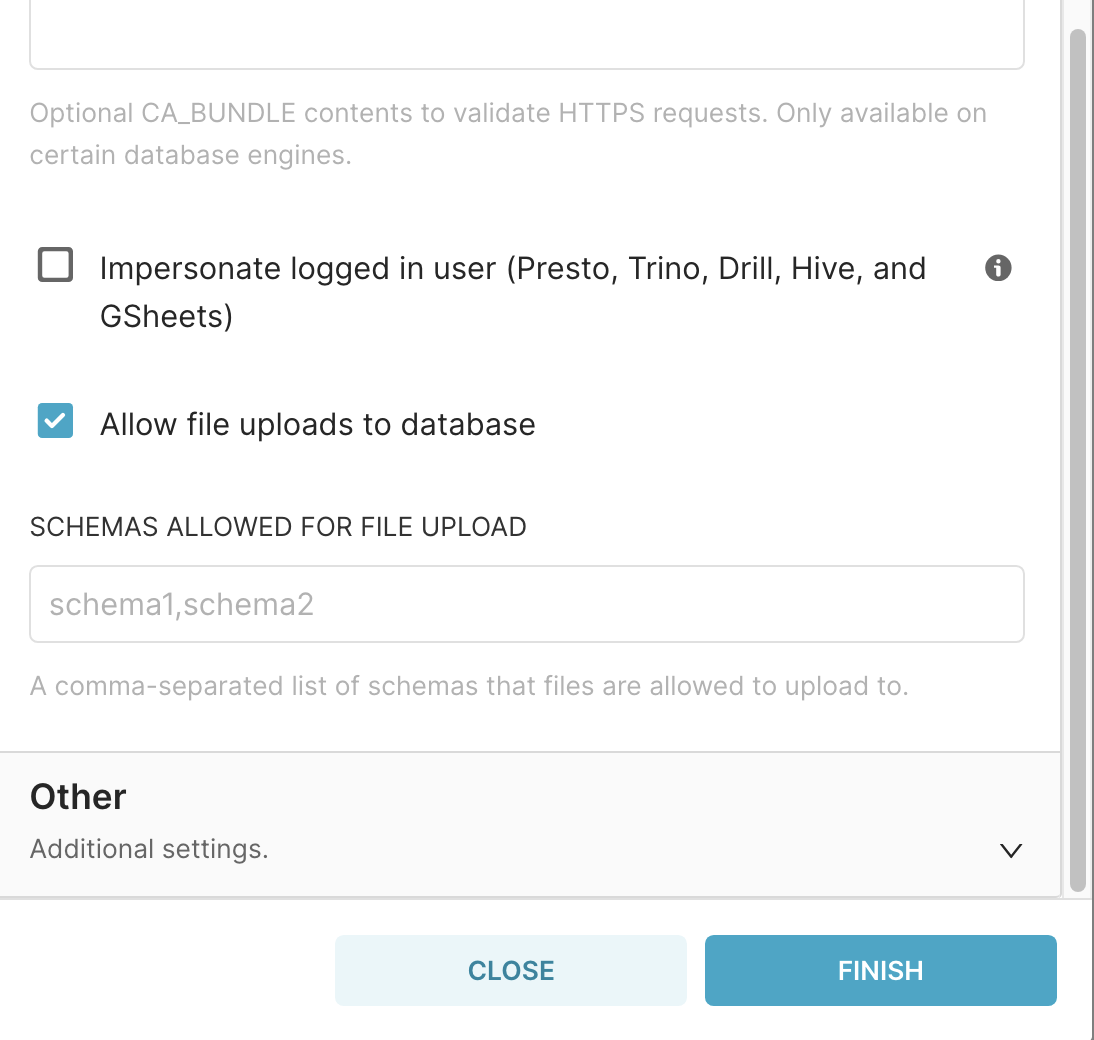
Выбрать examples - Edit



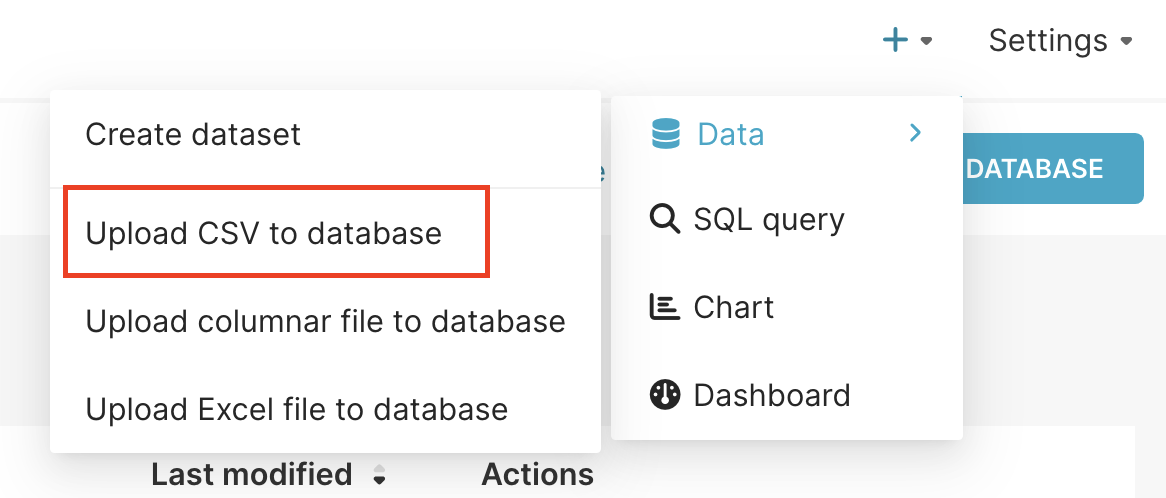


 Перейти в Security:

Поставить галку Allow file uploads



И тогда будет доступна опция:



# **Установка Postgres (по желанию)**

Создать папку и в ней файл с содержимым:

Открыть терминал, перейти в эту папку:

cd /Volumes/Data/docker/superset2

Создать файл docker-compose.yaml. с текстом:

version: '3.3'

services:

postgres:

image: library/postgres:13

volumes:

- postgres-data:/var/lib/postgresql/data

environment:

POSTGRES\_USER: postgres

POSTGRES\_HOST\_AUTH\_METHOD: trust

ports:

- '5432:5432'

restart: always

networks:

- pg-net

pgadmin:

image: dpage/pgadmin4

volumes:

- pgadmin-data:/var/lib/pgadmin

ports:

- '5050:80'

environment:

PGADMIN\_DEFAULT\_EMAIL: postgres@pg.com

PGADMIN\_DEFAULT\_PASSWORD: xxx

depends\_on:

- postgres

networks:

- pg-net

volumes:

postgres-data:

pgadmin-data:

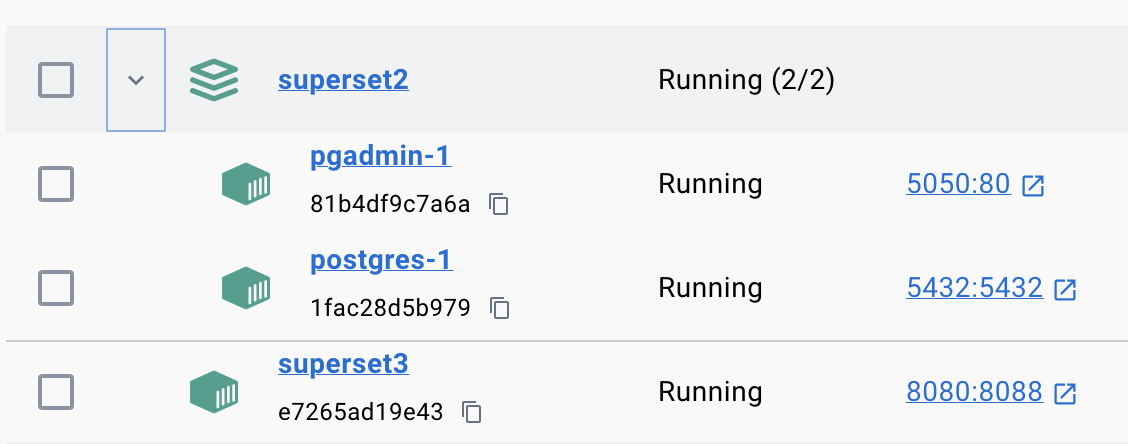
networks:

pg-net:

Выполнить команду:

docker compose up -d

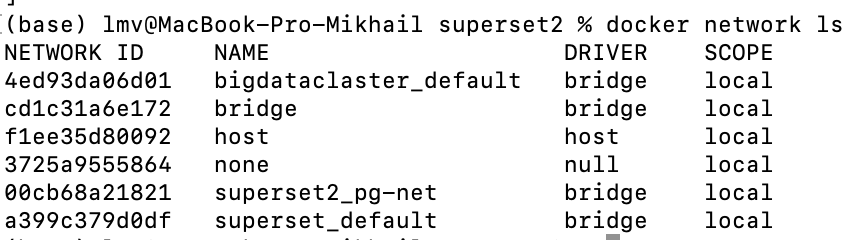
Будут развернуты 2 новых контейнера с базой данных и консолью для администрирования



Необходимо объединить их в одной сети:

Посмотреть список сетей:

docker network ls



Сеть с базой данных:

superset2\_pg-net

Команда покажет имена и IP адреса виртуалок:

docker inspect superset2\_pg-net

Команда перенесет контейнер **superset** в сеть **superset2\_pg-net**

docker network connect superset2\_pg-net superset

Проверить:

docker inspect superset2\_pg-net

"ConfigOnly": false,

"Containers": {

"1fac28d5b97921fbf25f60fd3e9bf252f7a2b868685e28c1934df17da2a45519": {

"Name": "superset2-postgres-1",

"EndpointID": "89471f42aad26ac012b4b9ebffae816966579378ed4dbc4bba5206d9dece97ad",

"MacAddress": "02:42:ac:18:00:02",

"IPv4Address": "172.24.0.2/16",

"IPv6Address": ""

},

"81b4df9c7a6a77fcf0cee9307e818c52020bb8d992a498978aed435910fdbb41": {

"Name": "superset2-pgadmin-1",

"EndpointID": "c4fc7204ca9b7fdfb498337b78518913f83999b05afa2cf40597307abfa64eef",

"MacAddress": "02:42:ac:18:00:03",

"IPv4Address": "172.24.0.3/16",

"IPv6Address": ""

},

"e7265ad19e43bd33e783e0028eb7484c6ba6609ffcd4780bd90a6be0241cc903": {

"Name": "superset3",

"EndpointID": "4e4b04cd00151a002161f876ce3cd9e371b3d21b4ecb276a4e0e04f16081be31",

"MacAddress": "02:42:ac:18:00:04",

"IPv4Address": "172.24.0.4/16",

"IPv6Address": ""

}

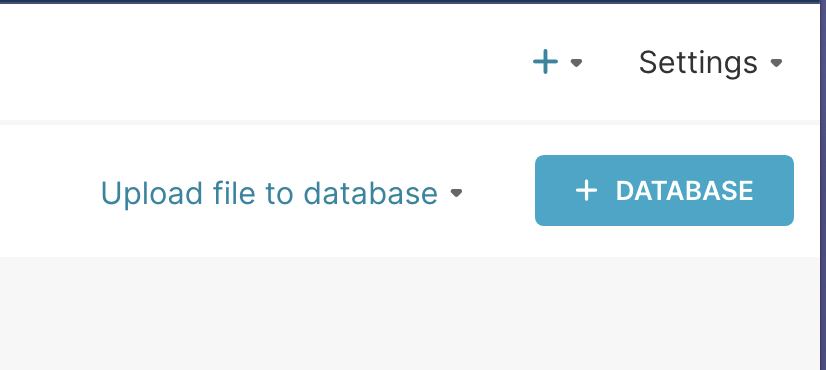
Все контейнеры в одной сети.

Посмотреть и скопировать адрес postgres - 172.24.0.2

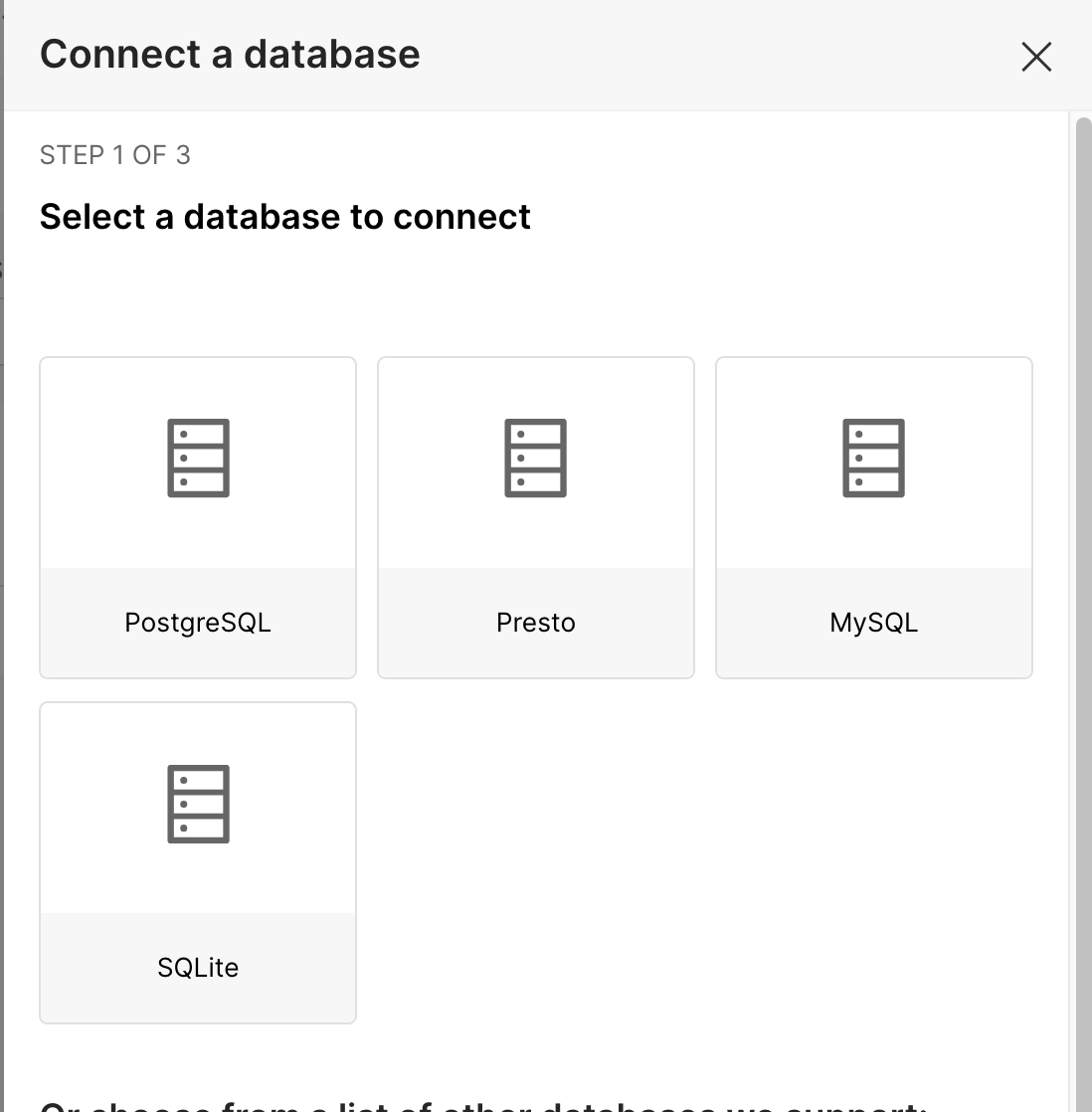
Перезагрузить  суперсет (в консоли Docker)

Перейти в Settings - DataBase Connections

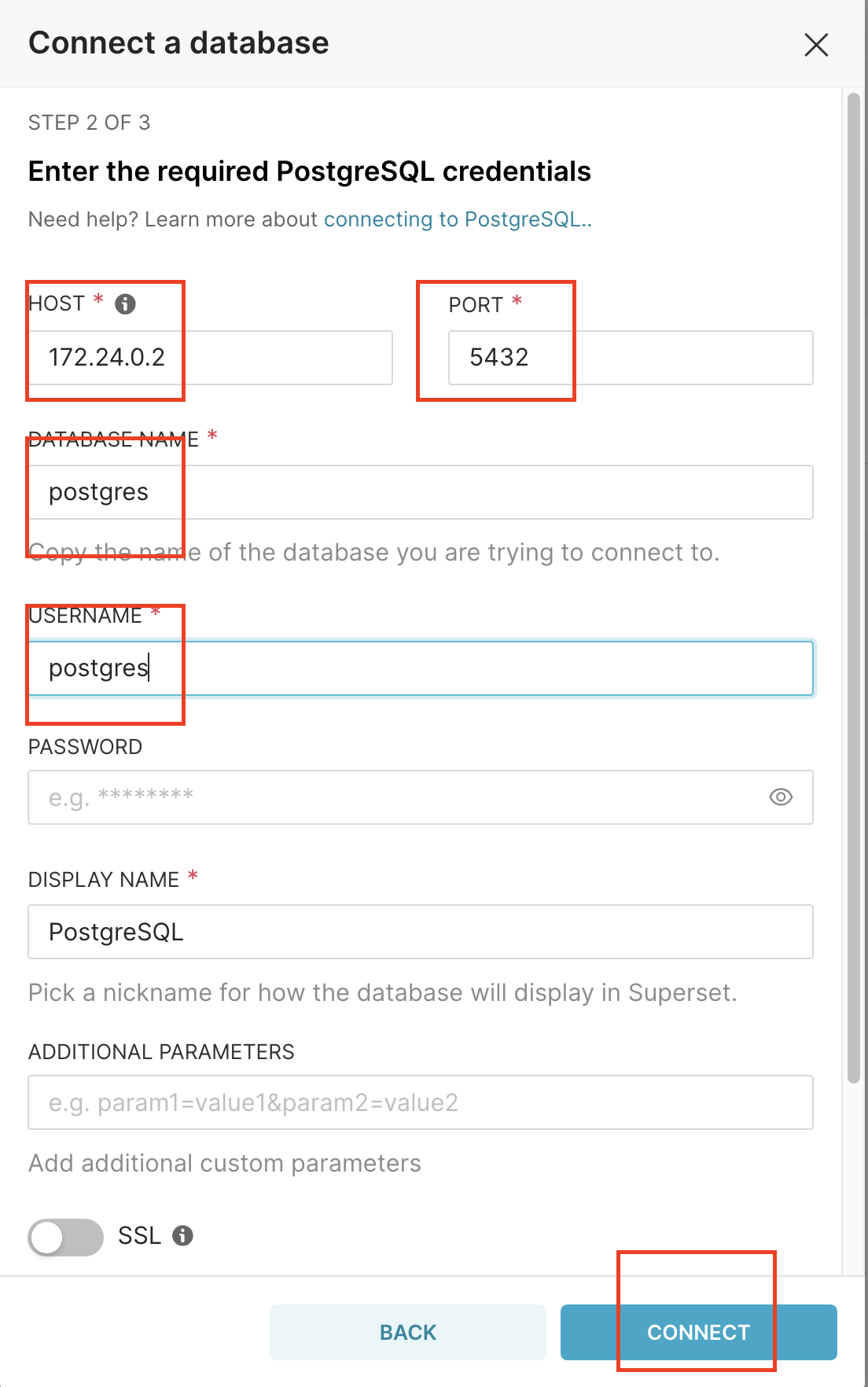
Нажать "+ Database"



В меню выбрать PostgreSQL



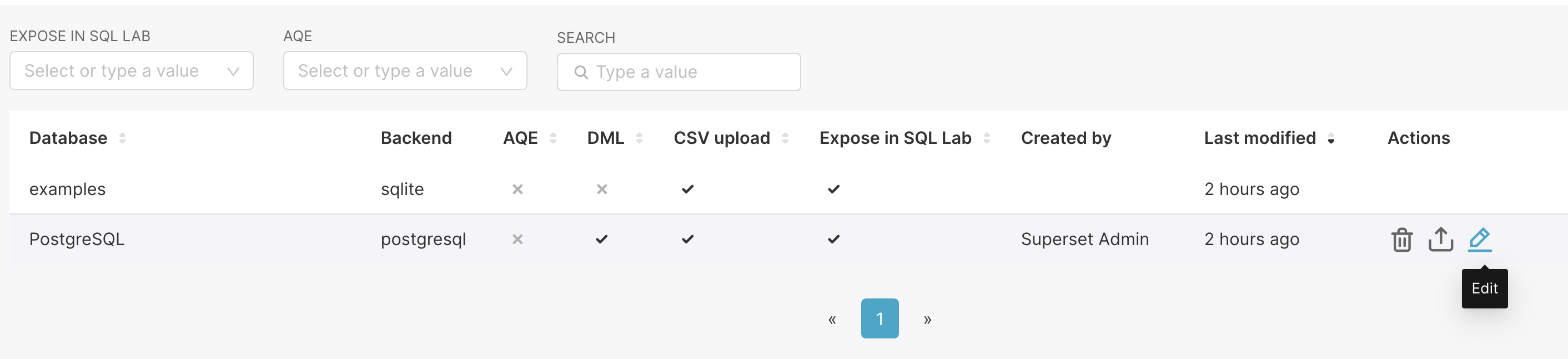
Ввести данные (ранее скопированный адрес сервера), пароль не вводить



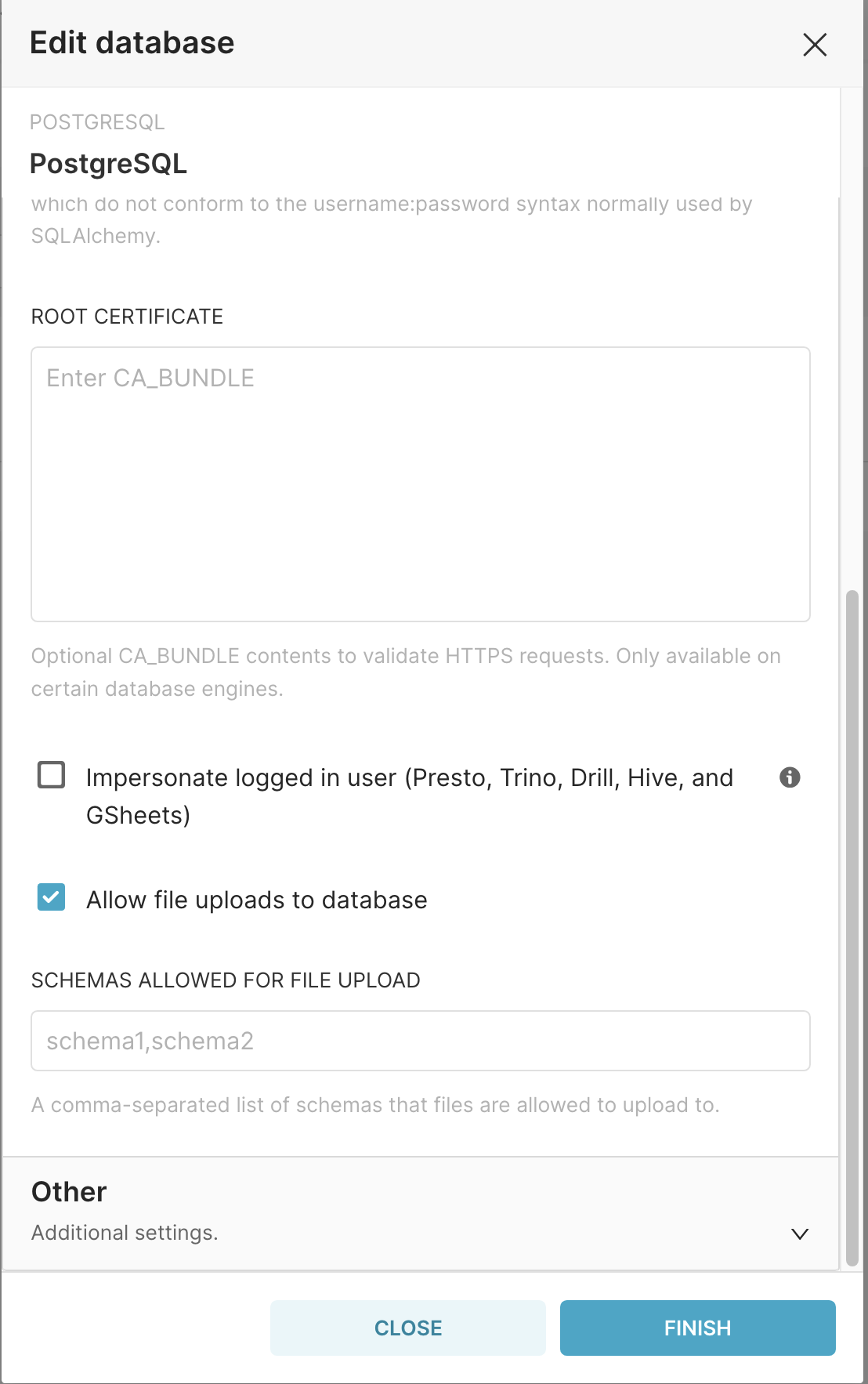
Перейти во вкладку ADVANSED

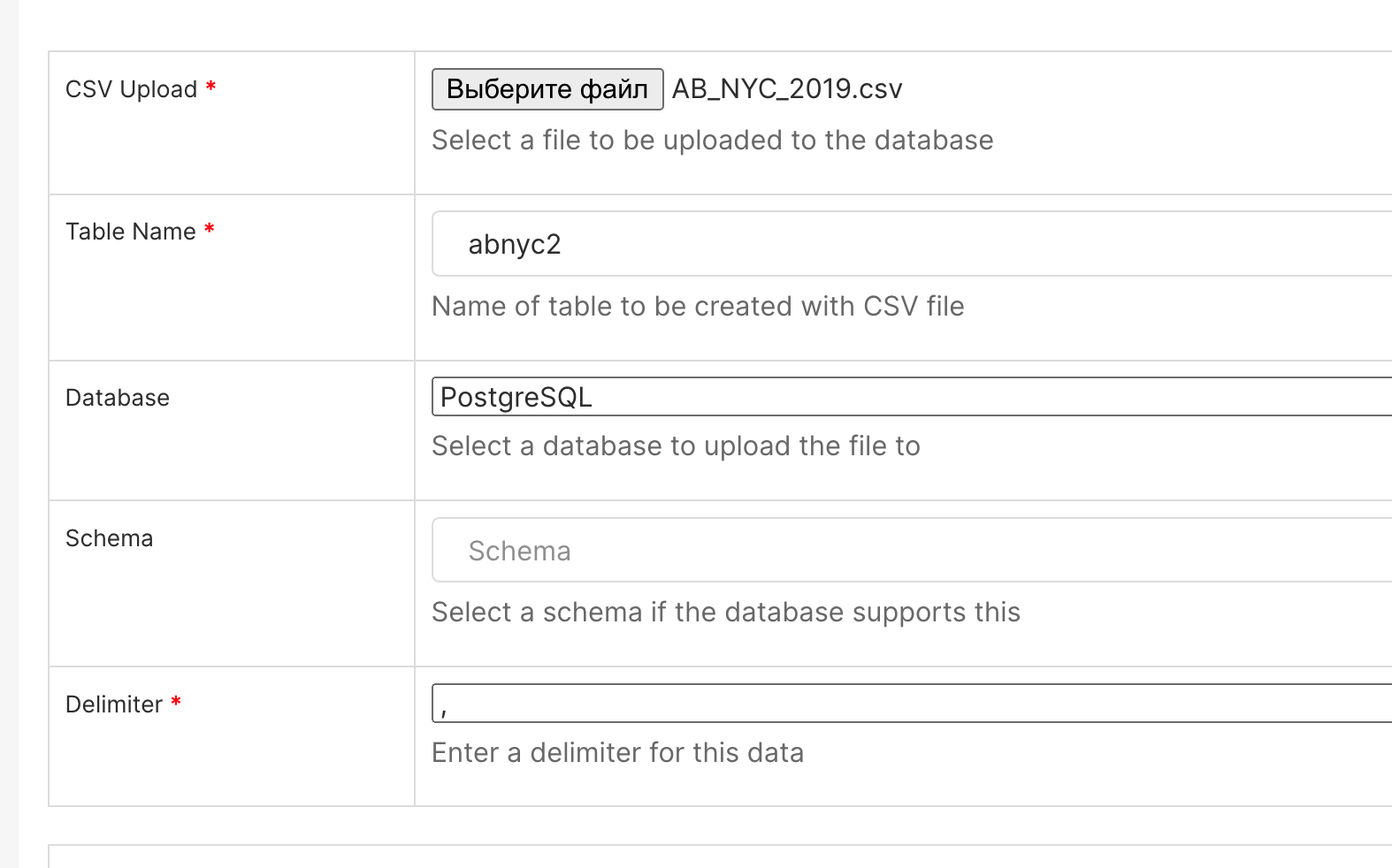


Появится новое подключение:



**В меню Edit - Advanced - поставить галку  Allow ...**



]

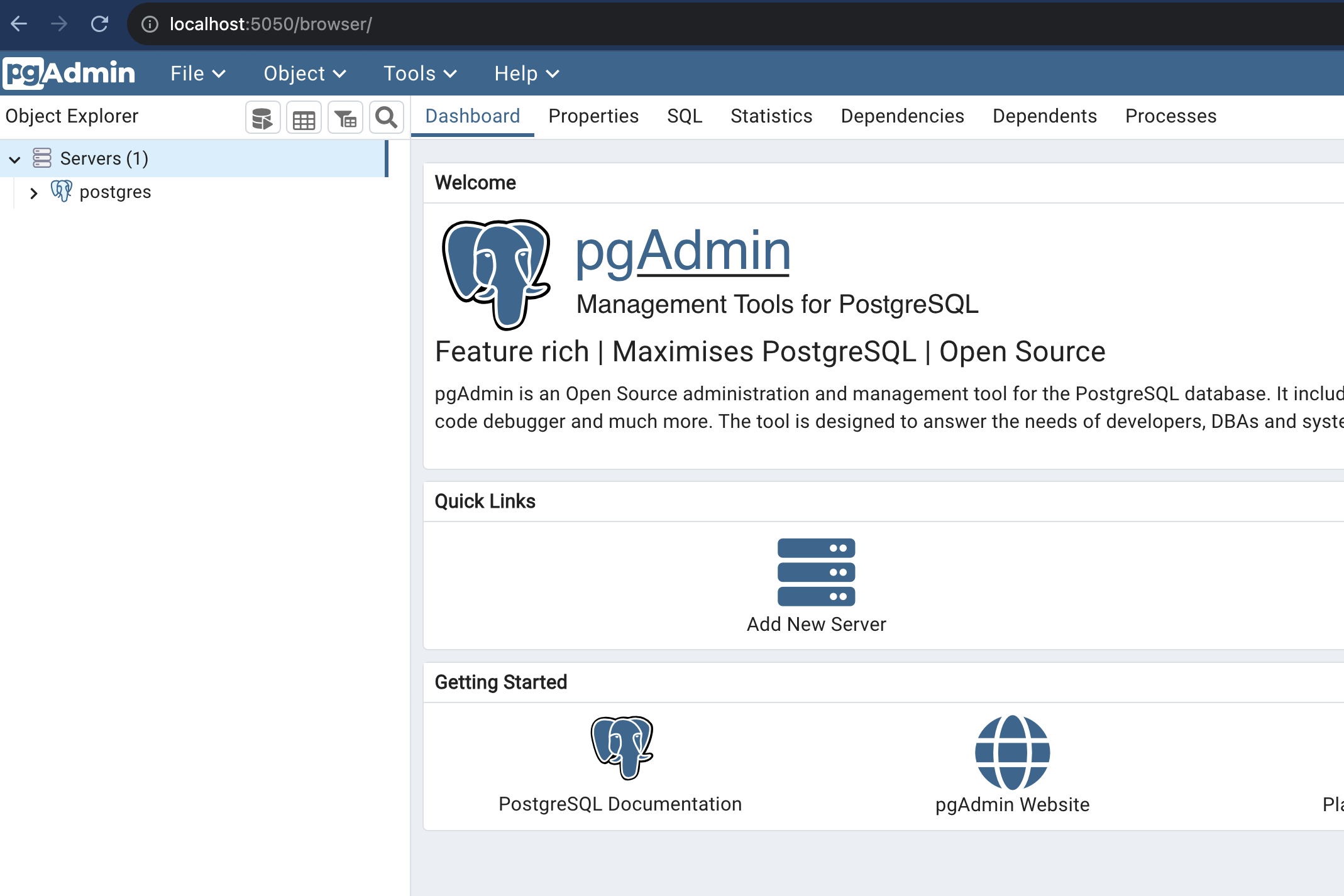
Можно так же настроить консоль для управления БД:

Перейти по ссылке:

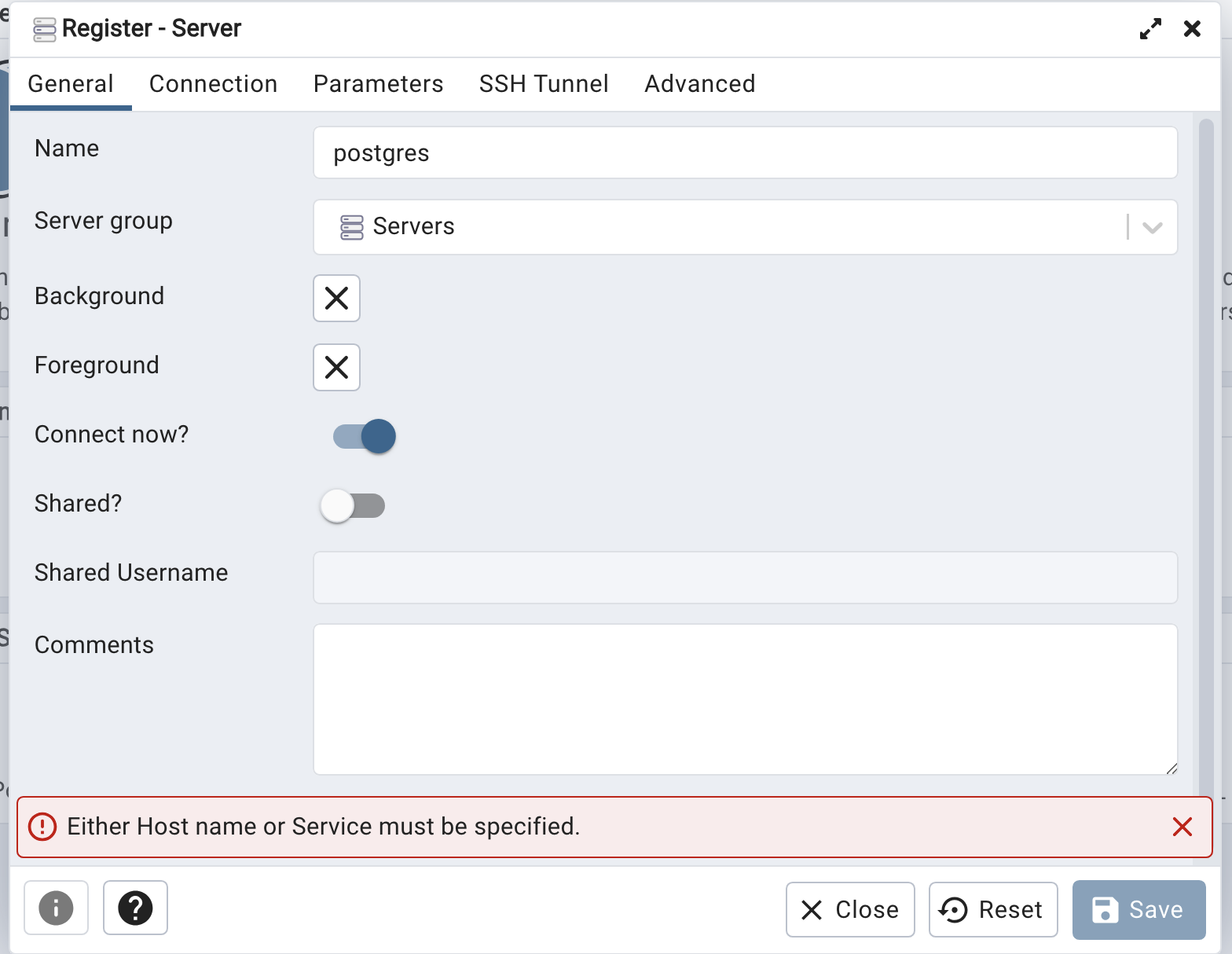
http://localhost:5050/browser/

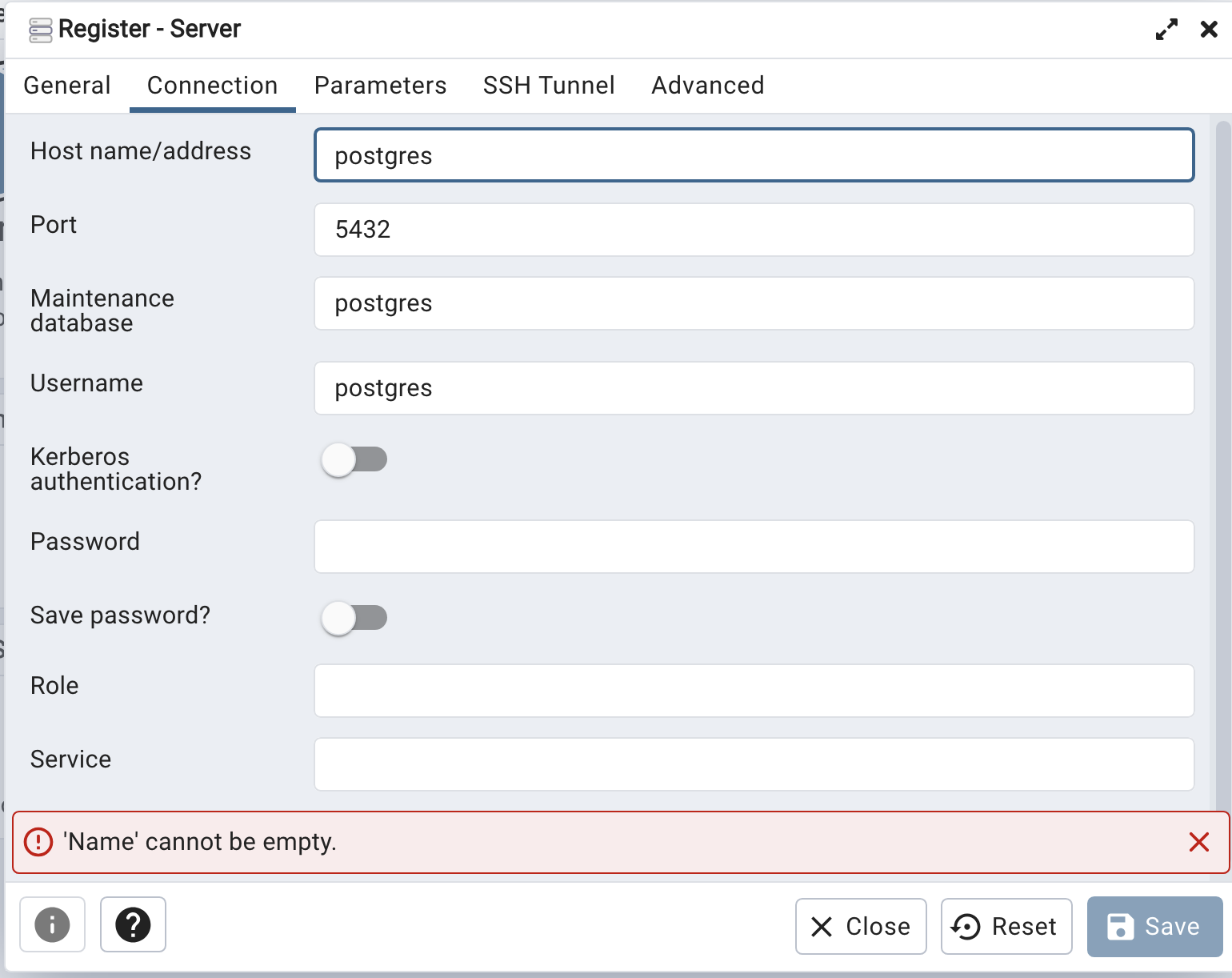
Логин: postgres@pg.com

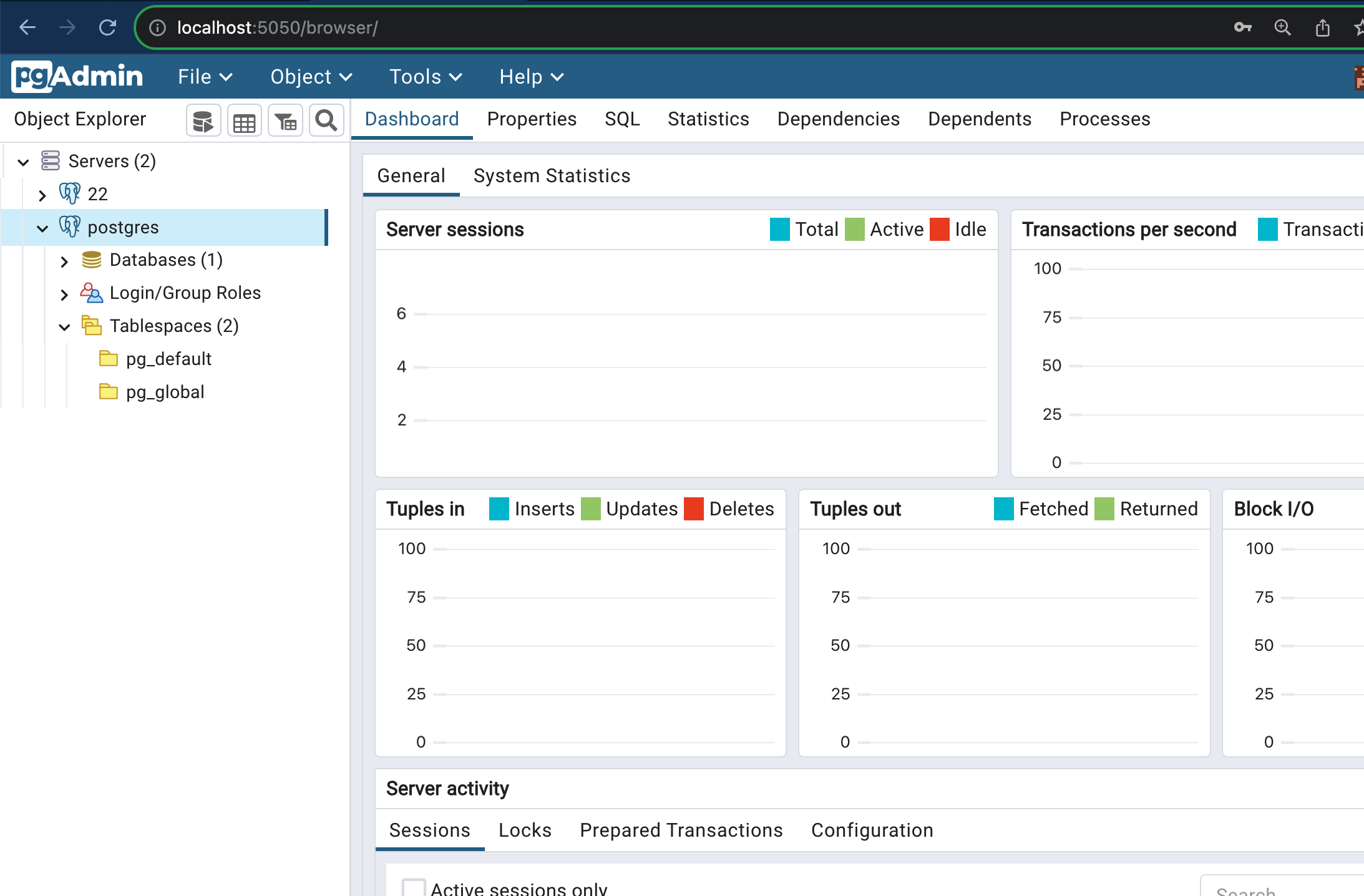
Пароль: xxx



Нажать **add new server и ввести необходимые параметры как на картинках**







# **ПРИЛОЖЕНИЕ**

Видеообзор инструкции:

<https://youtu.be/DmqFljyi5r0>

<https://t.me/LMV_DS>

Исходная инструкция:

<https://habr.com/ru/articles/661159/>

Дополнительная инструкция:

<https://superset-bi.ru/how-to-install-apache-superset-in-5-minutes-using-docker/>

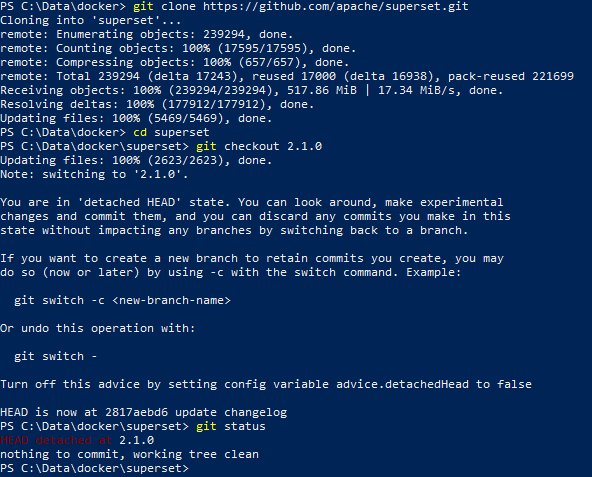
!curl -sSLo data.csv 'https://drive.google.com/uc?export=download&id=1g8lsZO25O2SxuCrIYVSDV3eLqmfC-TCZ'

# **Установка superset**

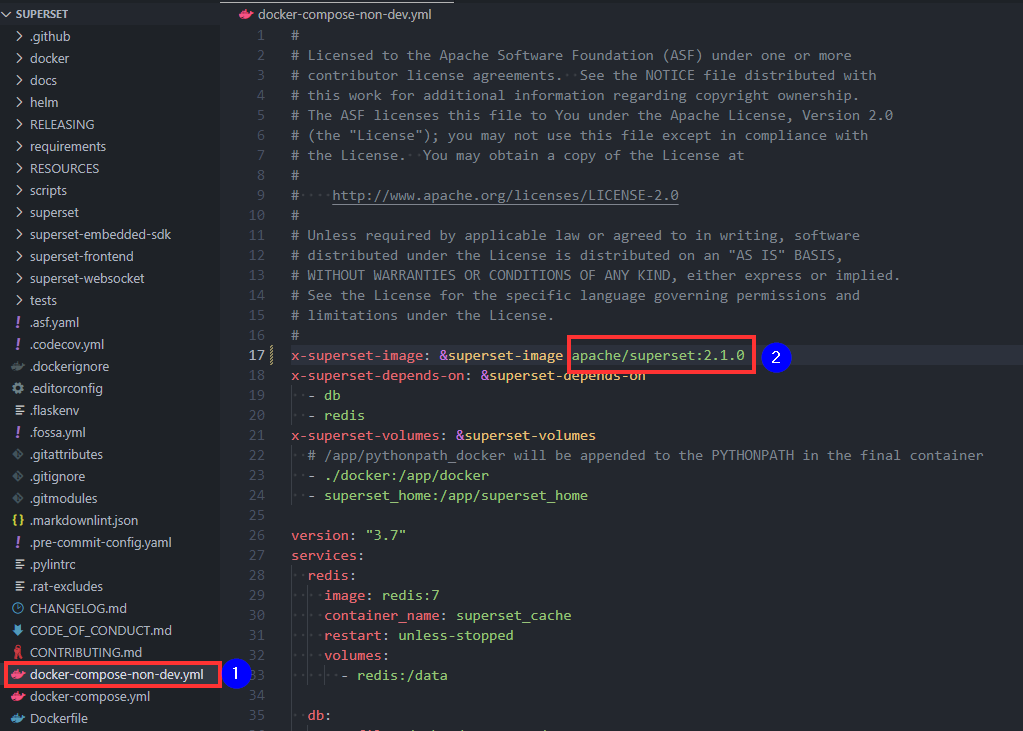
git clone <https://github.com/apache/superset.git>

cd superset

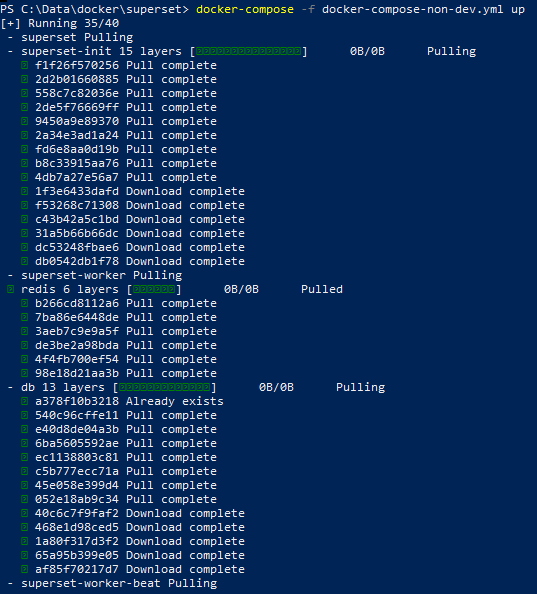
git checkout 2.1.0



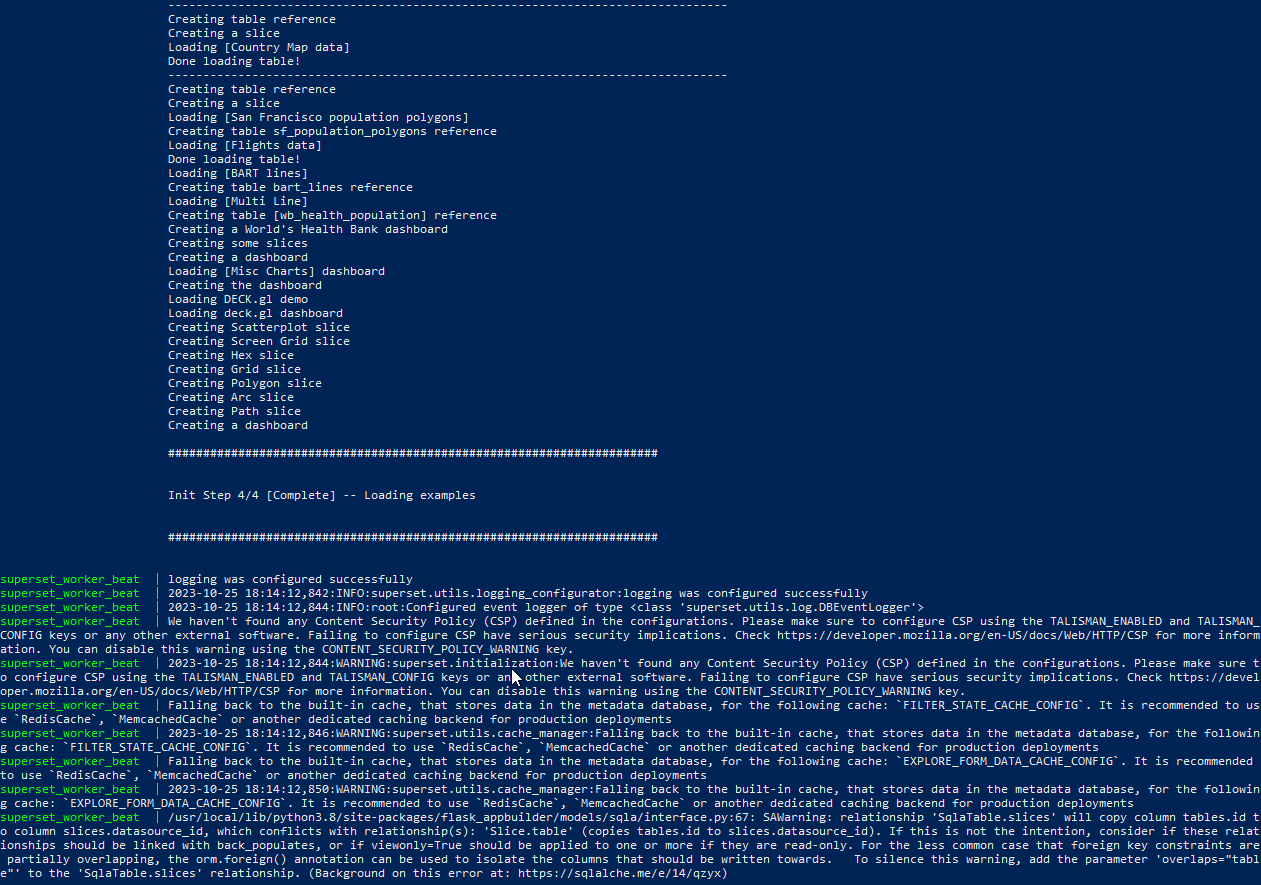
Изменить версию



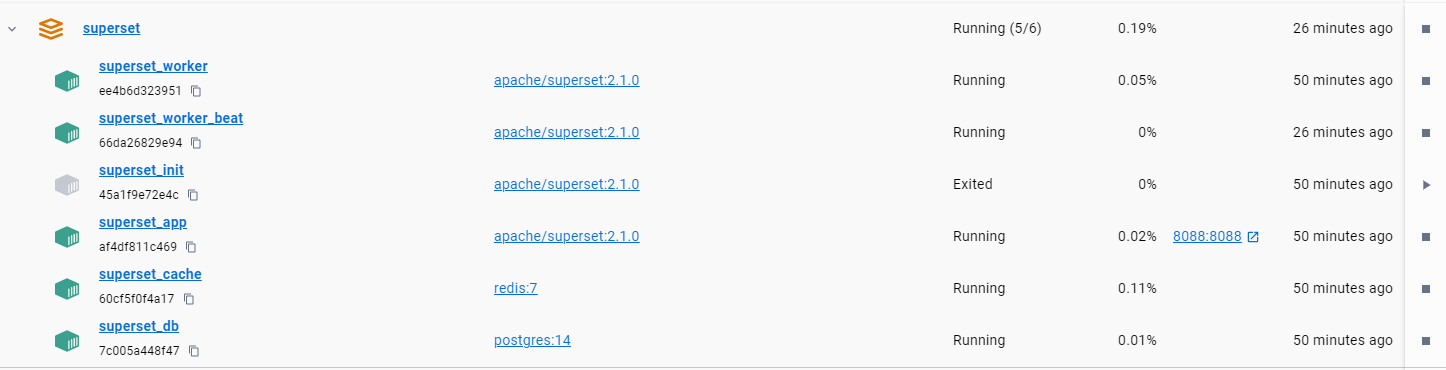
docker-compose -f docker-compose-non-dev.yml up



Загрузке примеров происходит в бесконечном цикле

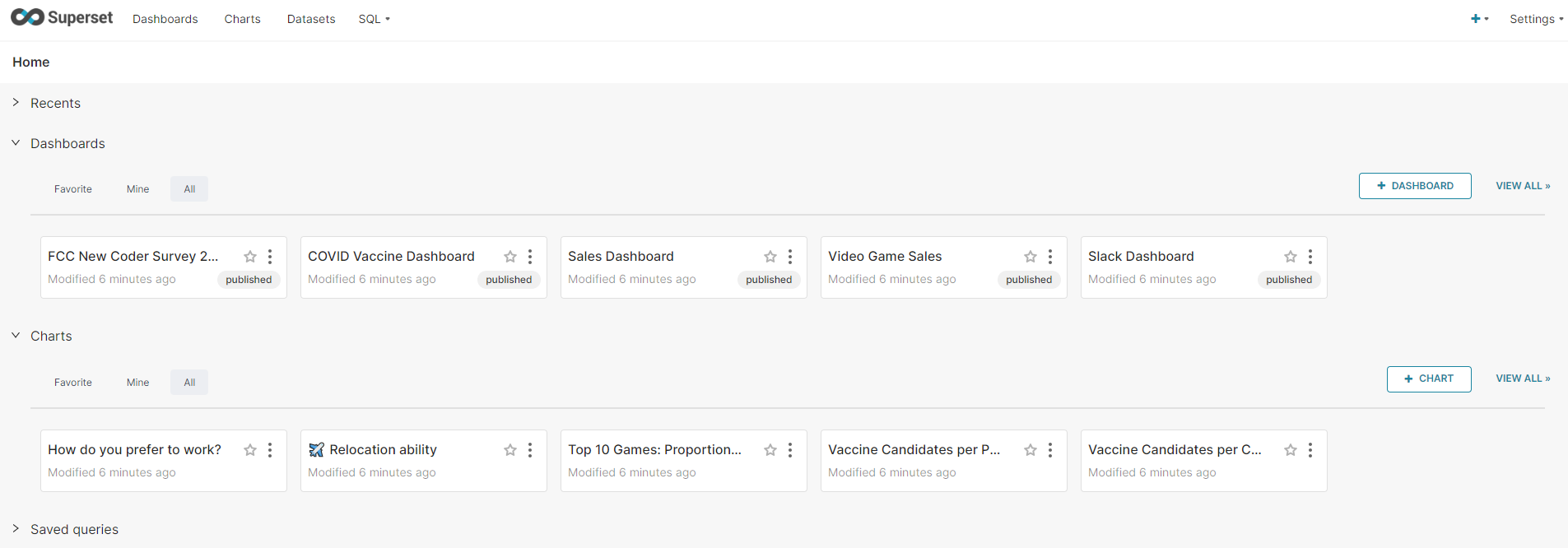


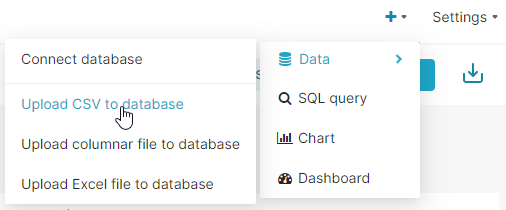
В итоге superset\_init не загружается



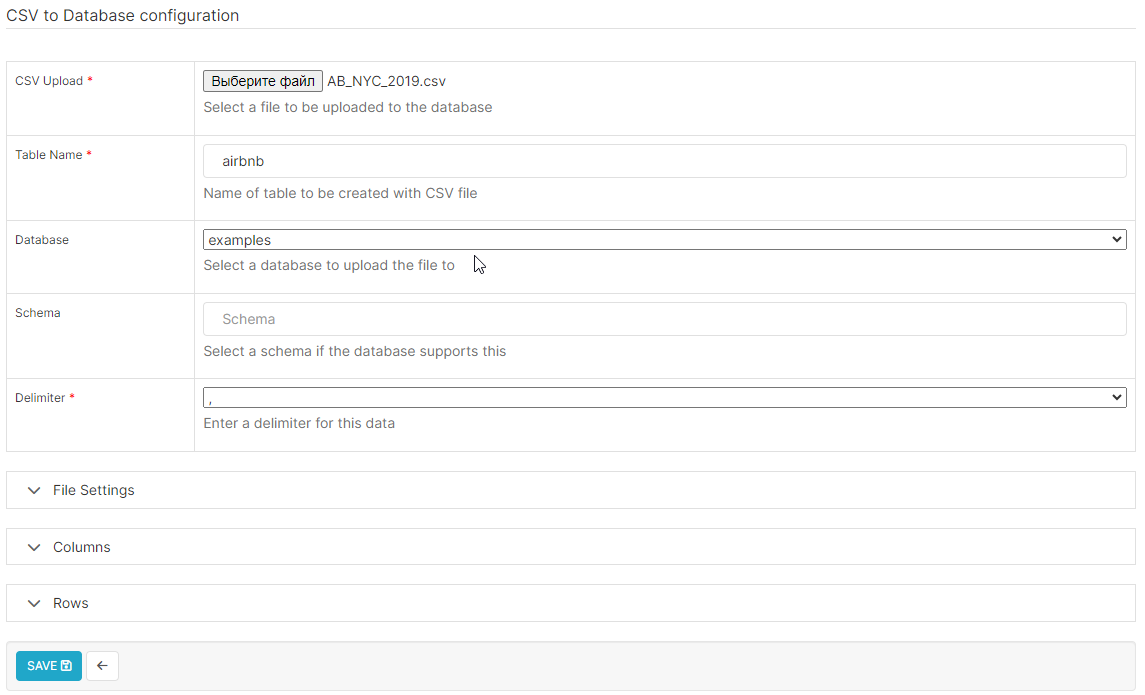
Тем не менее superset можно пользоваться

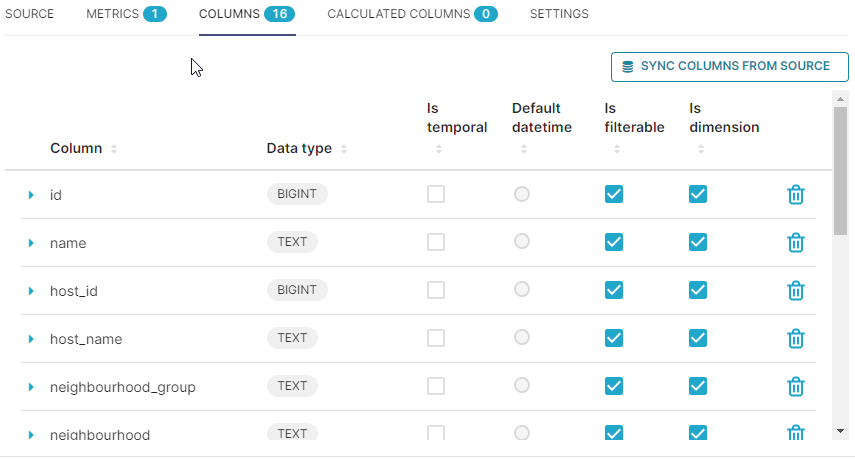
<http://localhost:8088>

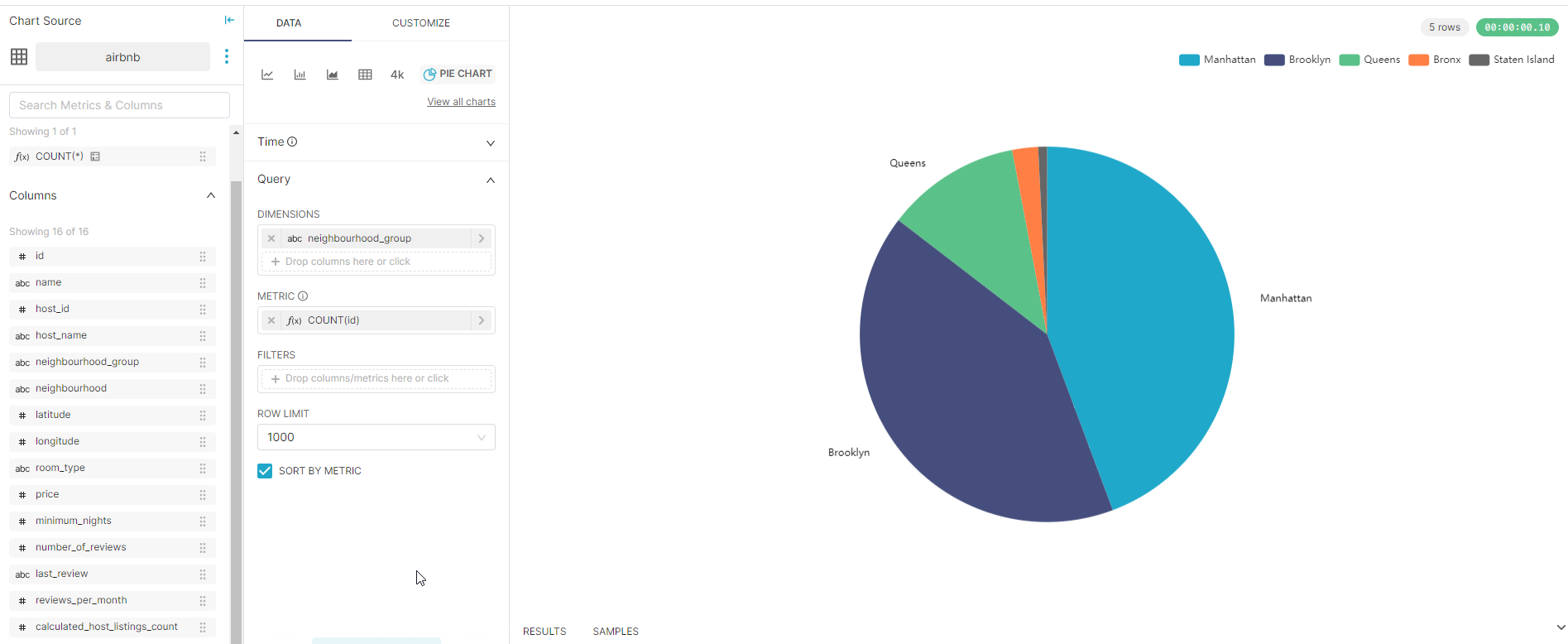


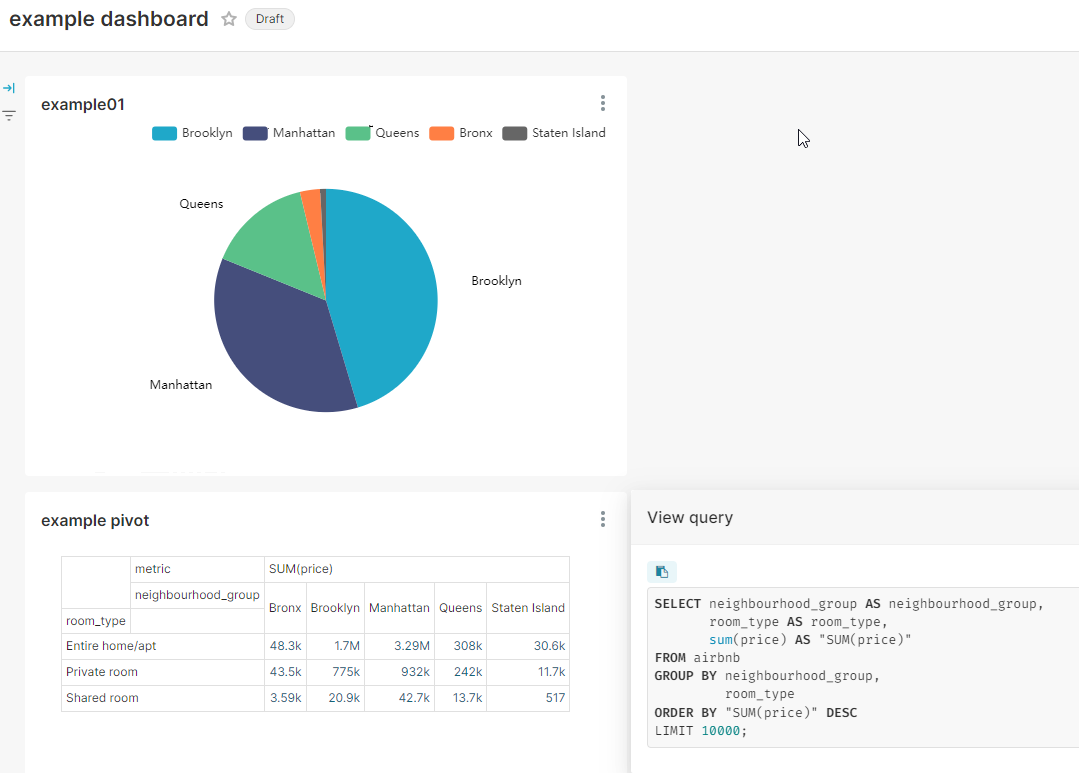


Загружаю для примера airbnb таблицу









# **IMPALA**

docker pull apache/impala:4.0.0-impala\_quickstart\_hms

docker run -d --name impala -p 21000:21000 -p 21050:21050 -p 25000:25000 -p 25010:25010 -p 25020:25020 apache/impala:4.0.0-impala\_quickstart\_hms impala

WORK

docker pull apache/kudu:impala-latest

docker run -d --name kudu-impala -p 21000:21000 -p 21050:21050 -p 25000:25000 -p 25010:25010 -p 25020:25020 apache/kudu:impala-latest impala

docker exec -it kudu-impala impala-shell

impala://172.24.0.5:21050/db

# **VAR 3**

git checkout 3.0.1rc1

в файле docker/pythonpath\_dev/superset\_config.py

в самом низу добавить

SESSION\_COOKIE\_SAMESITE = None

SESSION\_COOKIE\_SECURE = False

SESSION\_COOKIE\_HTTPONLY = False

PREVIOUS\_SECRET\_KEY = 'ZLfWZYSYmB1LBDf16ozTBgDoAHVKWFgr96I5lAUElNPFeMDiU3USrQEn'

SECRET\_KEY = 'd2F/ndWp5hUo114PfI93CZL/17uCR11M7BEFMu9dFNKKwyE88eP1DZmo'

в файле docker-compose-non-dev.yml

x-superset-image: &superset-image apachesuperset.docker.scarf.sh/apache/superset:3.0.1rc1 вместо latest

TAG=3.0.1rc1 docker-compose -f docker-compose-non-dev.yml up -d

Это все актуально для версии 3.0.1rc1