



# Docker | Dbeaver

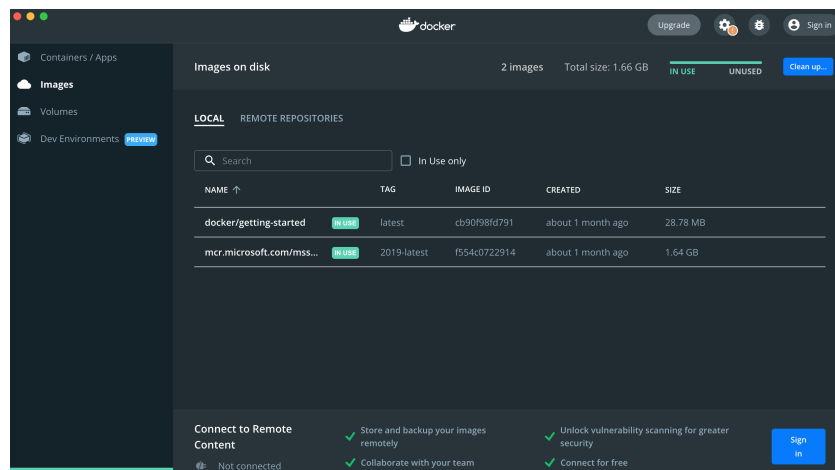
## Docker

Скачиваем **Docker**: [Ссылка на скачивание](#)

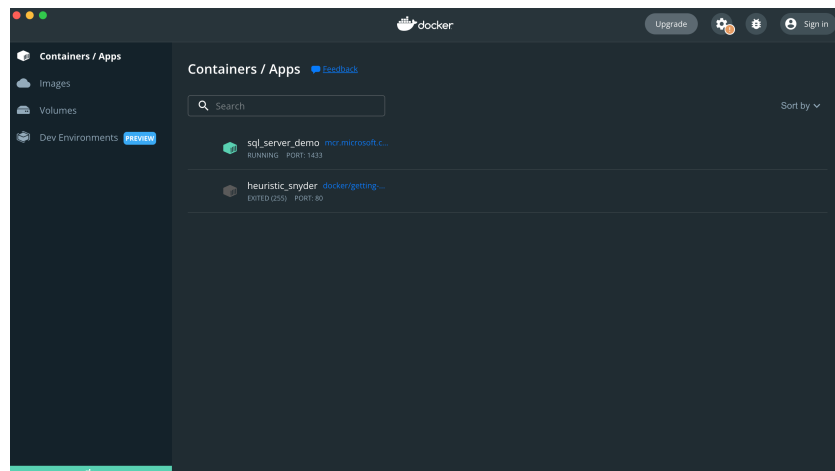
## Подключаемся к MS SQL Server в Docker-е

### Instruction Docker

```
# Command to pull MSSQL Docker Image #Образ  
sudo docker pull mcr.microsoft.com/mssql/server:2019-latest
```



```
# Command to run the container #Контейнер  
docker run -d --name sql_server_demo -e 'ACCEPT_EULA=Y' -e 'SA_PASSWORD=reallyStrongPwd123' -p 1433:1433 mcr.microsoft.com/mssql/server:2019-latest
```

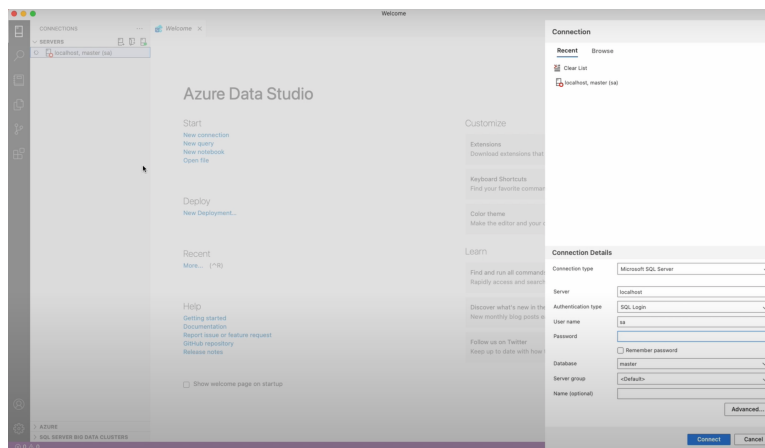
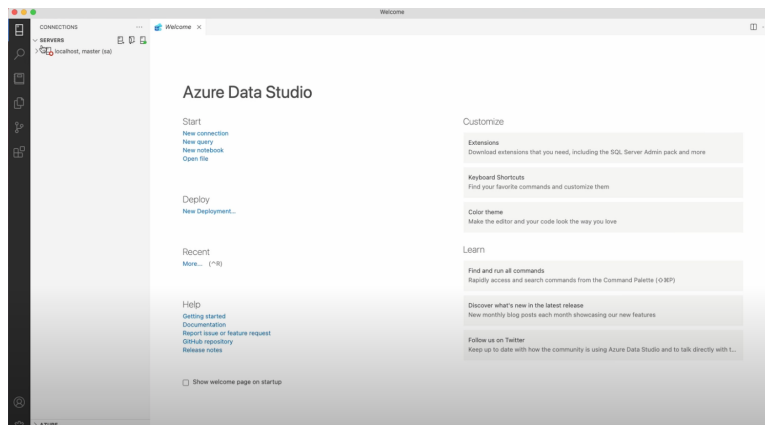


```
# Установка sql-cli
npm install -g sql-cli #ИЛИ
sudo npm install -g sql-cli

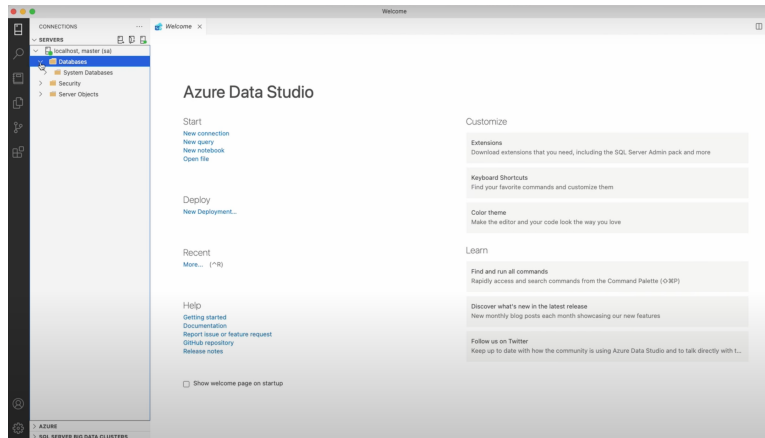
#Если выдает ошибку нужно установить npm
#mac OS with brew:
brew install the nodeJS
```

## Azure

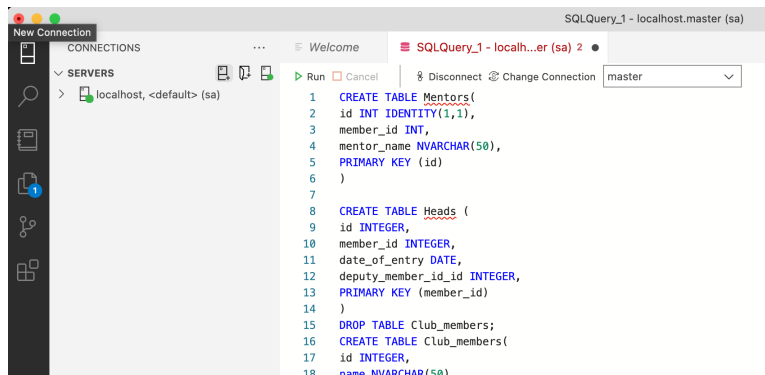
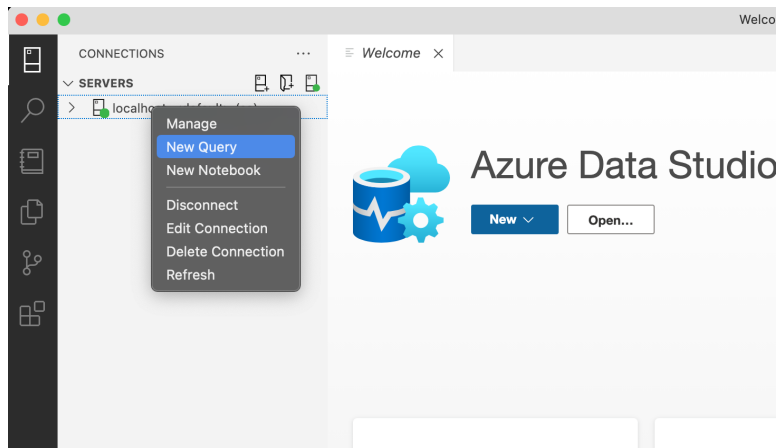
Скачиваем Azure: [Ссылка на скачивание](#)



В строке Password используем пароль, который использовали ранее для поднятия контейнера в Docker:  
**reallyStrongPwd123**



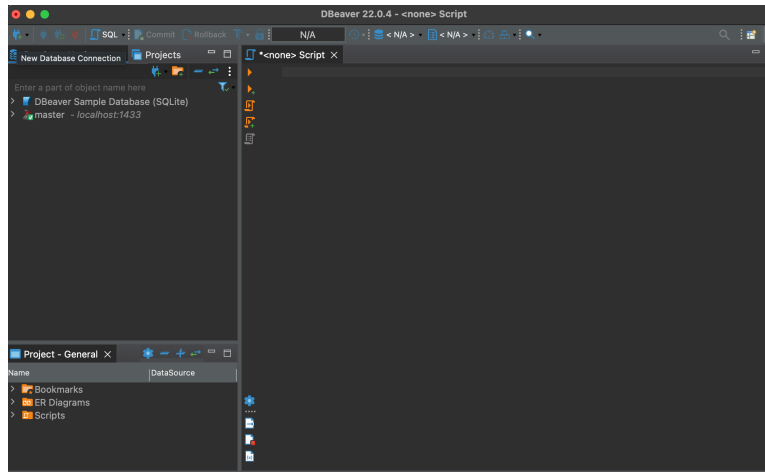
Чтобы запустить скрипт нажимаем левой кнопкой мыши на localhost и выбираем New Query. В новом окошке вставляем скрипт, который хотим запустить и нажимаем на Run



## Dbeaver

Скачиваем Dbeaver: [Ссылка на скачивание](#)

Открываем приложение



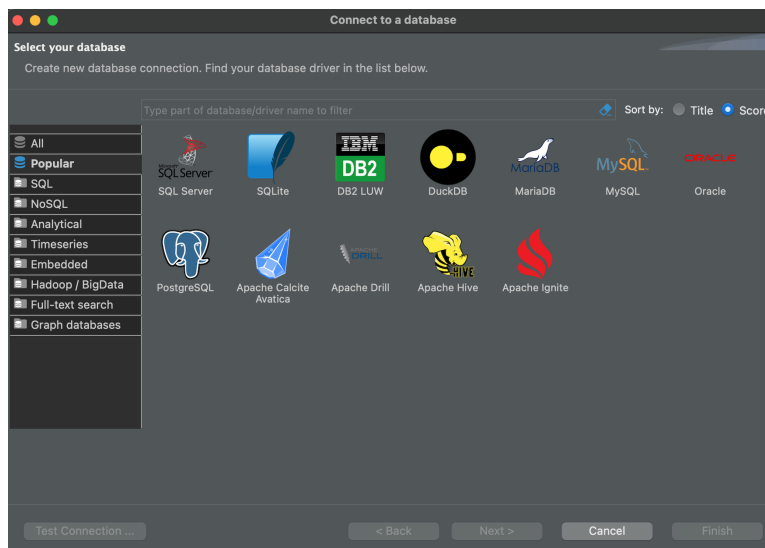
В левом верхнем углу розетка, нажимаем на нее

В открытом окне выбираем **SQL Server** и нажимаем Next >

UserName: sa

Password: reallyStrongPwd123

**Готово, поздравляю!**



Connect to a database

### SQL Server Connection Settings

MS SQL Server / SQL Server connection settings

Microsoft SQL Server

Main Driver properties SSH Proxy SSL

Host: localhost Port: 1433

Database/Schema: master

Authentication

Authentication: SQL Server Authentication

Username:

Password: ☒ Save password locally

Settings

☐ Show All Schemas

*You can use variables in connection parameters.* Connection details (name, type, ...)

Driver name: MS SQL Server / SQL Server Edit Driver Settings

Test Connection ... < Back Next > Cancel Finish