

## 1. Zainstaluj docker desktop for windows

<https://hub.docker.com/editions/community/docker-ce-desktop-windows>

## 2. Uruchom infrastrukturę

2.1. Przejdź do folderu z repozytorium kodu, a następnie przejdź do folderu **Compose**

2.2. Uruchom skrypt **start\_infr.bat**

```
D:\BalticLSC\source\baltilsc-backend\Compose>start_infr.bat

D:\BalticLSC\source\baltilsc-backend\Compose>docker stack deploy --compose-file=infrastructure.yml baltic
Creating network baltic
Creating config baltic_rabbitmq_conf
Creating config baltic_consul_postgresql
Creating config baltic_consul_json
Creating config baltic_enabled_plugins
Creating service baltic_postgres
Creating service baltic_pgadmin
Creating service baltic_consul
Creating service baltic_fabio
Creating service baltic_rabbitmq

D:\BalticLSC\source\baltilsc-backend\Compose>docker stack services baltic
ID                NAME                MODE                REPLICAS            IMAGE                PORTS
b55ml4lg8b4e      baltic_consul        replicated           0/1                  consul:latest        *:8400->8400/tcp, *:8500->8500/tcp, *:8502->8502/tcp, *:8600->53/tcp
p7wl3r9w77o3      baltic_rabbitmq      replicated           0/1                  rabbitmq:latest      *:5672->5672/tcp, *:15672->15672/tcp
tacg84od4ajy      baltic_postgres      replicated           0/1                  postgres:latest      *:5432->5432/tcp
vrvtxr1fue4s      baltic_pgadmin        replicated           0/1                  dpag/pgadmin4:latest *:5050->80/tcp
zloh8u0l7scp      baltic_fabio         replicated           0/1                  fabio/fabio:latest   *:9998-9999->9998-9999/tcp

D:\BalticLSC\source\baltilsc-backend\Compose>docker volume ls
DRIVER    VOLUME NAME
local     app_rgb2edge_swarm_infr_mongoconfig
local     app_rgb2edge_swarm_infr_mongodb
local     baltic_consul
local     baltic_pgadmin
local     baltic_postgres
local     portainer_data

D:\BalticLSC\source\baltilsc-backend\Compose>
```

2.3. Poczeka chwilę i uruchom skrypt **status.bat**

```
D:\BalticLSC\source\baltilsc-backend\Compose>docker stack services baltic
ID                NAME                MODE                REPLICAS            IMAGE                PORTS
b55ml4lg8b4e      baltic_consul        replicated           1/1                  consul:latest        *:8400->8400/tcp, *:8500->8500/tcp, *:8502->8502/tcp, *:8600->53/tcp
p7wl3r9w77o3      baltic_rabbitmq      replicated           1/1                  rabbitmq:latest      *:5672->5672/tcp, *:15672->15672/tcp
tacg84od4ajy      baltic_postgres      replicated           1/1                  postgres:latest      *:5432->5432/tcp
vrvtxr1fue4s      baltic_pgadmin        replicated           1/1                  dpag/pgadmin4:latest *:5050->80/tcp
zloh8u0l7scp      baltic_fabio         replicated           1/1                  fabio/fabio:latest   *:9998-9999->9998-9999/tcp

D:\BalticLSC\source\baltilsc-backend\Compose>
```

1/1 oznacza, że kontener został uruchomiony, nie oznacza to, że aplikacja wewnątrz kontenera została już uruchomiona (pgAdmin uruchamia się najdłużej)

Dostęp do WebUI poszczególnych serwisów:

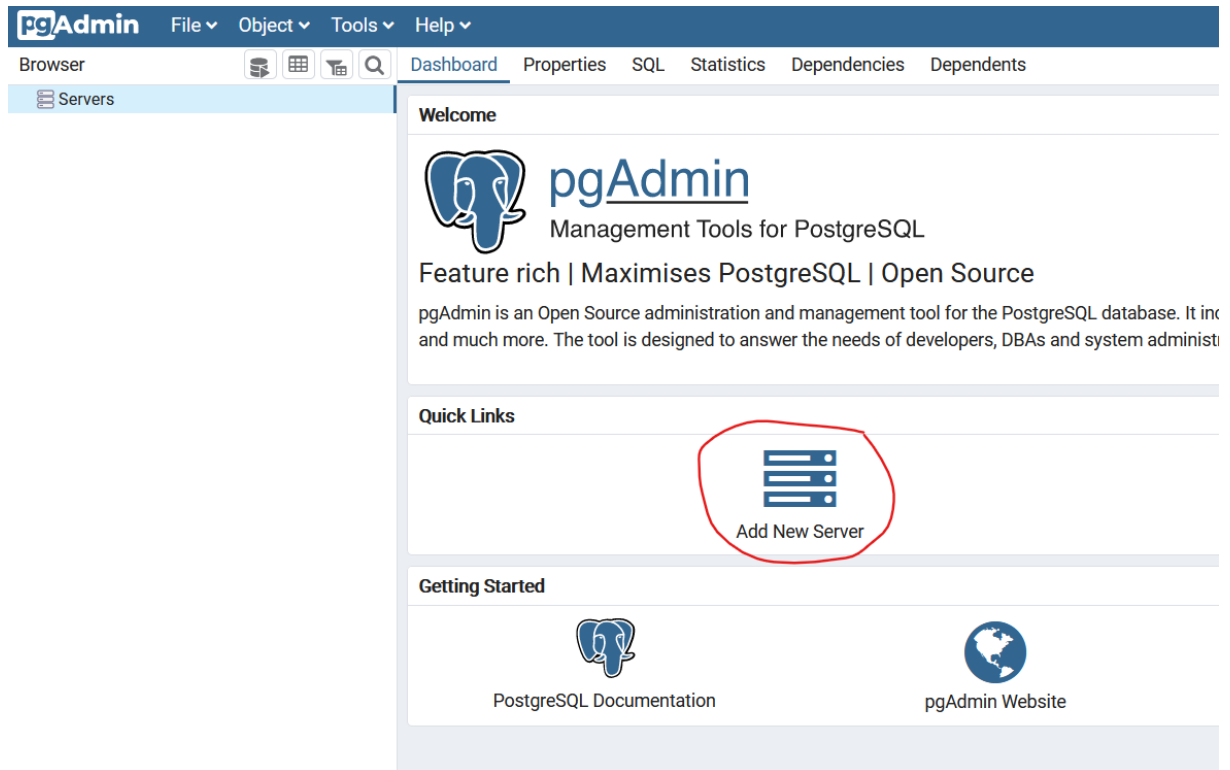
- Consul: <http://localhost:8500>
- RabbitMQ: <http://localhost:15672>
- pgAdmin: <http://localhost:5050>
- fabio: <http://localhost:9998>

Dane logowania do panelów administracyjnych, są w pliku infrastructure.yml lub jeżeli nie zostały ustawione, to używane są standardowe (do znalezienia na stronie z opisem kontenera)

3. Stwórz bazę danych **baltic** (jednorazowo chyba, że wykonano polecenie **docker system prune**)

3.1. Otwórz panel administracyjny pgAdmin <http://localhost:5050>

3.2. Dodaj nowy serwer



3.2.1. Zakładka General

### 3.2.2. Zakładka Connection

Create - Server

General **Connection** SSL SSH Tunnel Advanced

Host name/address: postgres

Port: 5432

Maintenance database: postgres

Username: postgres

Password: P@ssw0rd

Save password? ☐

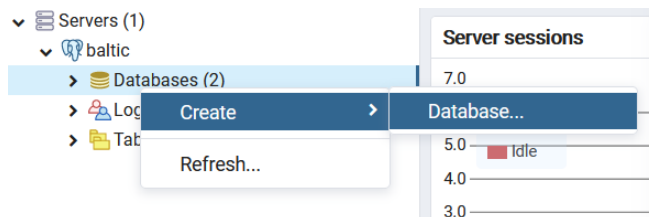
Role:

Service:

Cancel Reset Save

### 3.3. Dodaj nową bazę

#### 3.3.1. Prawy klik myszy na Database



#### 3.3.2. Zakładka General

Create - Database

General Definition Security Parameters SQL

Database: baltic

Owner: postgres

Comment:

Cancel Reset Save

### 4. Dodaj sekret do projektu (tylko raz przed pierwszym uruchomieniem projektu)

#### 4.1. Uruchom Ridera oraz zainstaluj plugin do obsługi sekretów

<https://plugins.jetbrains.com/plugin/10183--net-core-user-secrets>

#### 4.2. Uruchom konsolę cmd lub power shell

#### 4.3. Przejdź do folderu projektu, który chcesz uruchomić

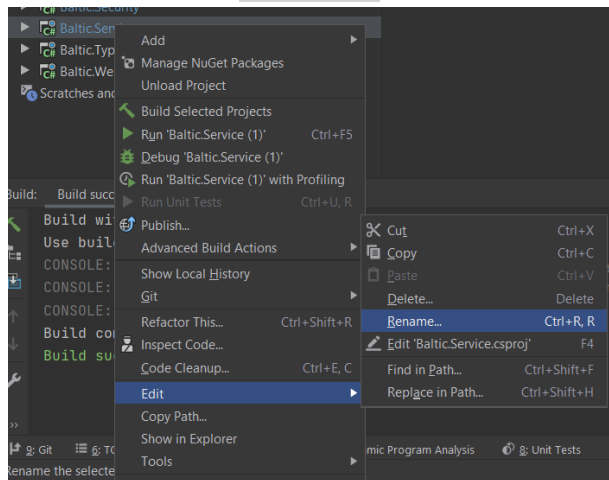
#### 4.4. Wygeneruj identyfikator użytkownika dla sekretu:

```
dotnet user-secrets init --project=Baltic.Server.csproj.user
```

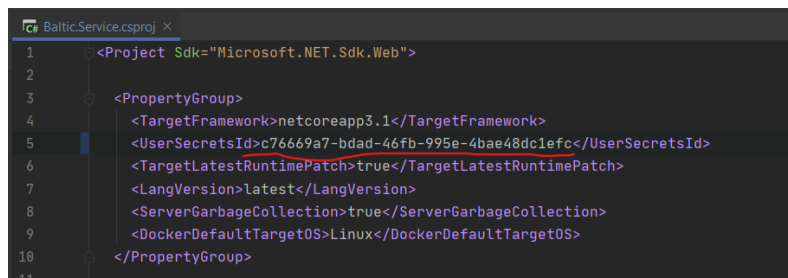
```
D:\BalticLSC\source\balticlsc-backend\Baltic.Server>dotnet user-secrets init --project=Baltic.Server.csproj.user
Set UserSecretsId to 'c7669a7-bdad-46fb-995e-4bae48dc1efc' for MSBuild project 'D:\BalticLSC\source\balticlsc-backend\Baltic.Server\Baltic.Server.csproj.user'.
```

#### 4.5. Dodaj sekret do projektu w Riderze

##### 4.5.1. Zmodyfikuj plik \*.cproj danego projektu



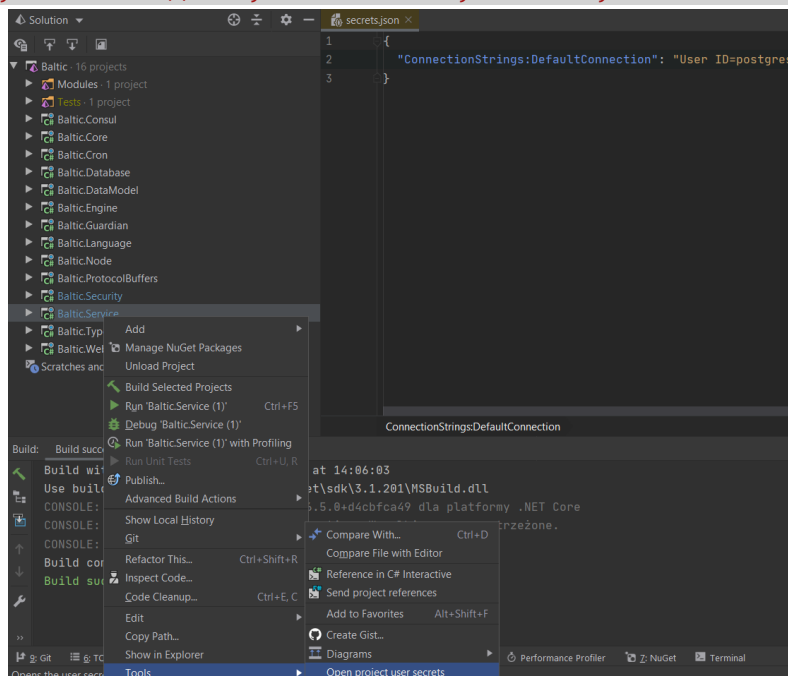
##### 4.5.2. Przekopiuuj wygenerowany identyfikator



##### 4.5.3. Uzupełnij zawartość sekretu o connectionString dla bazy

"ConnectionStrings:DefaultConnection": "User

ID=postgres;Password=Pa\$\$w0rd;Host=localhost;Port=5432;Database=baltic;Pooling=true;"



W celu uruchomienia następnego projektu wystarczy dodać identyfikator sekretu do pliku \*.cproj (Identyfikator wskazuje na zewnętrzny plik)

5. Gotowe, można pracować nad zmianami