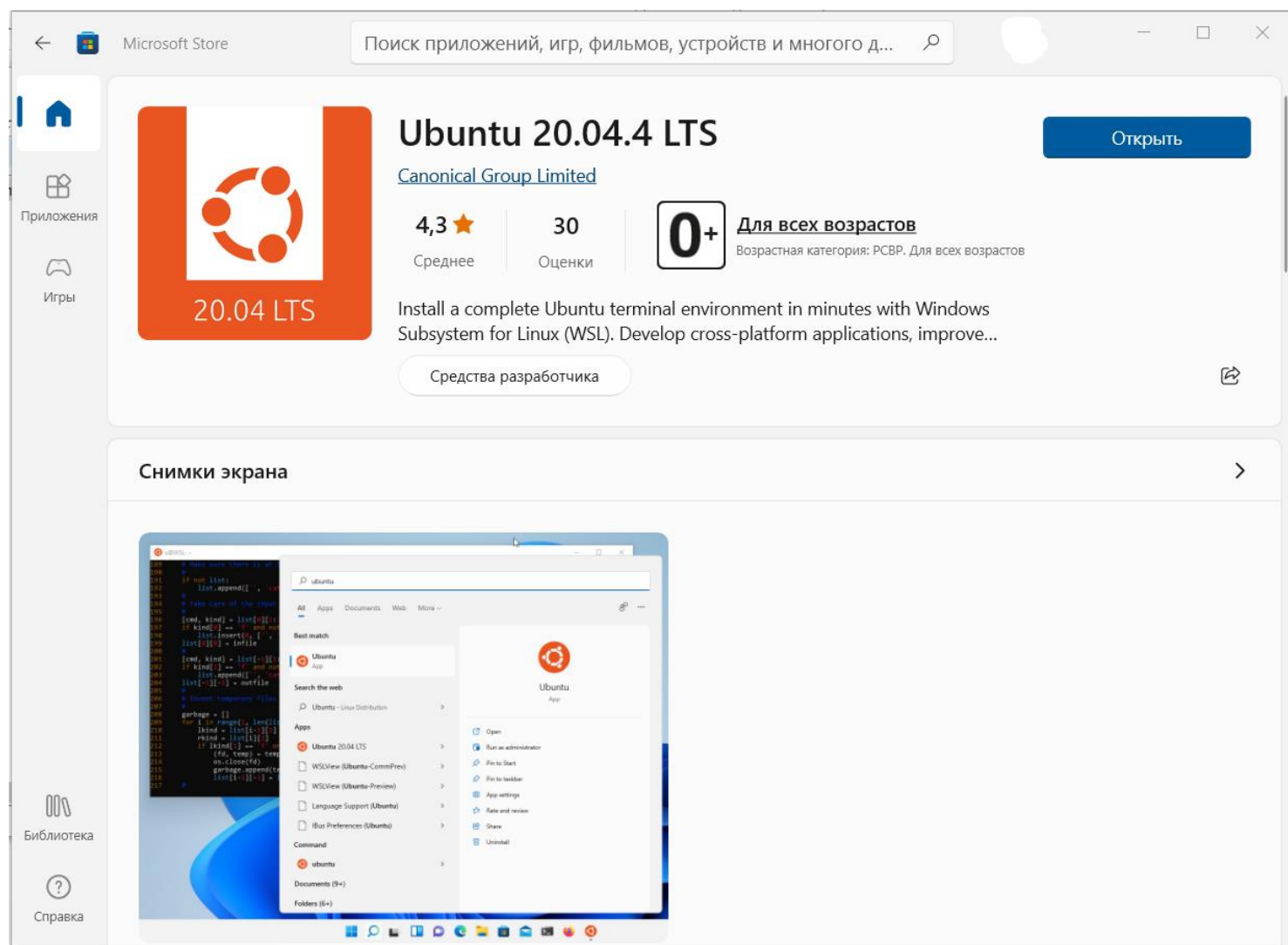


В рамках выполнения домашнего задания на машине с установленной Windows 10 PRO x64 было сделано следующее:

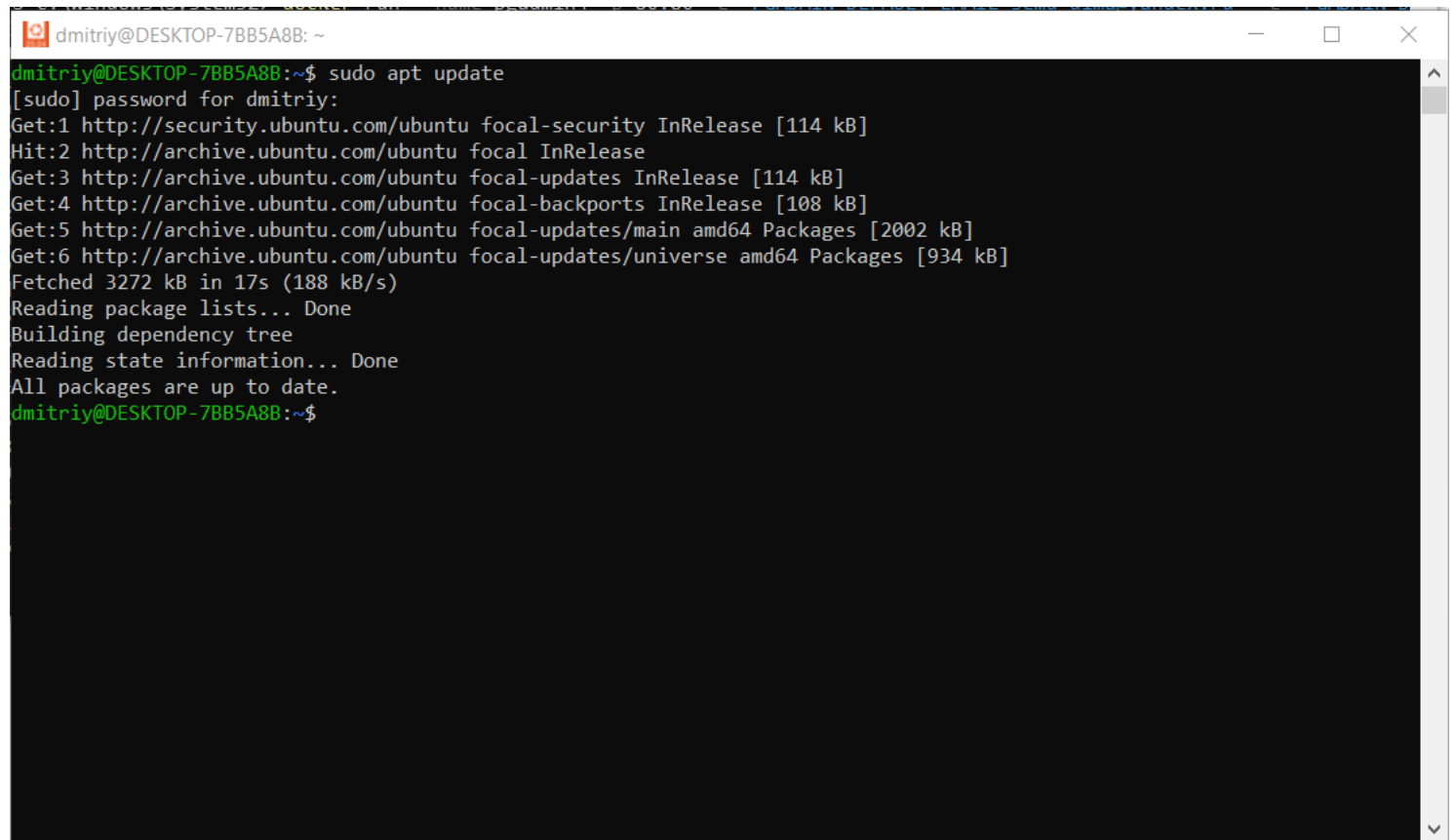
- клиентская оболочка с графическим интерфейсом pgadmin4 была запущена в программном обеспечении для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации, контейнеризатор приложений, Docker;
- сервер баз данных PostgreSQL был развернут в ПО WSL 2. WSL 2 — это новая версия архитектуры подсистемы Windows для Linux, которая поддерживает подсистему Windows для Linux, чтобы запускать двоичные файлы Linux ELF64 в Windows. В WSL 2 применялся дистрибутив GNU/Linux, скачанный с магазина приложений Microsoft Store, Ubuntu 20.04.4 LTS.

Находим, скачиваем и устанавливаем из магазина приложений Microsoft Store дистрибутив Ubuntu 20.04.4 LTS



После установки, запускаем WSL2 и вводим команду:

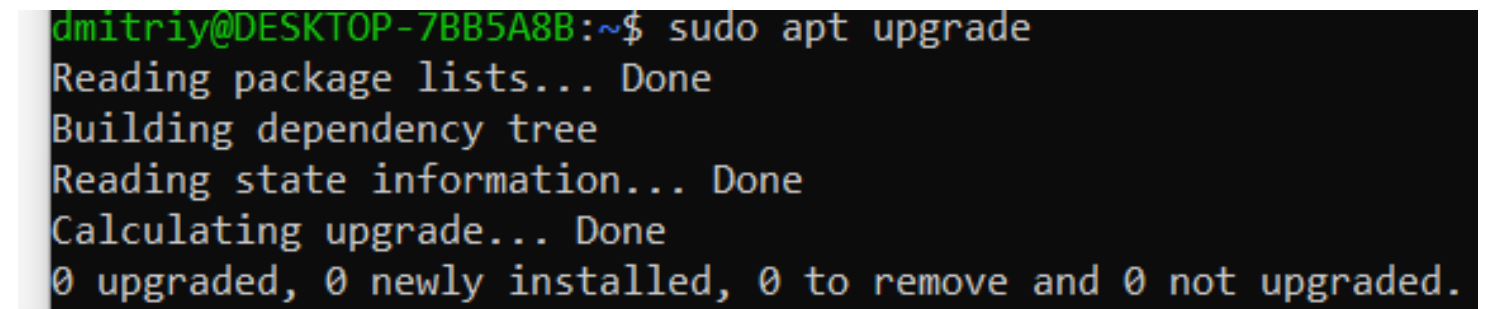
\$ sudo apt update

A terminal window titled 'dmitriy@DESKTOP-7BB5A8B: ~' with standard window controls. The terminal shows the execution of 'sudo apt update'. It prompts for a password, then lists several updates from Ubuntu repositories, including security updates and package updates for focal, focal-updates, focal-backports, and focal-updates/universe. It reports that 3272 kB were fetched in 17 seconds at 188 kB/s. Finally, it states that all packages are up to date.

```
dmitriy@DESKTOP-7BB5A8B:~$ sudo apt update
[sudo] password for dmitriy:
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Hit:2 http://archive.ubuntu.com/ubuntu focal InRelease
Get:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:5 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [2002 kB]
Get:6 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [934 kB]
Fetched 3272 kB in 17s (188 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
dmitriy@DESKTOP-7BB5A8B:~$
```

Вводим команду:

\$ sudo apt upgrade

A terminal window showing the execution of 'sudo apt upgrade'. It reports that package lists are read, the dependency tree is built, and state information is read. It then states that no upgrades are available, as all packages are already at the latest version.

```
dmitriy@DESKTOP-7BB5A8B:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

Командой:

```
$ sudo apt install postgresql postgresql-contrib
```

скачиваем и устанавливаем сервер базы данных postgresql

```
dmitriy@DESKTOP-7BB5A8B:~$ sudo apt install postgresql postgresql-contrib
Reading package lists... Done
Building dependency tree
Reading state information... Done
postgresql-contrib is already the newest version (12+214ubuntu0.1).
Suggested packages:
  postgresql-doc
The following NEW packages will be installed:
  postgresql
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 3924 B of archives.
After this operation, 67.6 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 postgresql all 12+214ubuntu0.1 [3924 B]
Fetched 3924 B in 0s (13.3 kB/s)
Selecting previously unselected package postgresql.
(Reading database ... 34350 files and directories currently installed.)
Preparing to unpack .../postgresql_12+214ubuntu0.1_all.deb ...
Unpacking postgresql (12+214ubuntu0.1) ...
Setting up postgresql (12+214ubuntu0.1) ...
```

После выполняем команды:

```
$ sudo apt update
```

```
$ sudo apt -y install postgresql-12 postgresql-client-12
```

```
dmitriy@DESKTOP-7BB5A8B:~$ sudo apt update
Hit:1 http://archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
dmitriy@DESKTOP-7BB5A8B:~$ sudo apt -y install postgresql-12 postgresql-client-12
Reading package lists... Done
Building dependency tree
Reading state information... Done
postgresql-12 is already the newest version (12.11-0ubuntu0.20.04.1).
postgresql-client-12 is already the newest version (12.11-0ubuntu0.20.04.1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

После завершения установки запустим службу postgresql командой:

```
$ sudo service postgresql start
```

и проверим ее состояние командой:

```
$ sudo service postgresql status
```

```
dmitriy@DESKTOP-7BB5A8B:~$ sudo service postgresql start
* Starting PostgreSQL 12 database server
dmitriy@DESKTOP-7BB5A8B:~$ sudo service postgresql status
12/main (port 5432): online
```

[OK]

Переключимся на пользователя postgres командой:

```
$ sudo su postgres
```

и командой

```
$ psql --version
```

проверим версию консольного клиента postgres

```
dmitriy@DESKTOP-7BB5A8B:~$ sudo su postgres
postgres@DESKTOP-7BB5A8B:/home/dmitriy$ psql --version
psql (PostgreSQL) 12.11 (Ubuntu 12.11-0ubuntu0.20.04.1)
```

Командой

```
$ psql
```

создадим консольного клиента psql



dmitriy@DESKTOP-7BB5A8B: ~

```
postgres@DESKTOP-7BB5A8B:/home/dmitriy$ psql
psql (12.11 (Ubuntu 12.11-0ubuntu0.20.04.1))
Type "help" for help.

postgres=#
```

Далее создаем пользователя:

```
CREATE USER dmitriy_s WITH PASSWORD 'security_pass';
```

создаем базу данных:

```
CREATE DATABASE test_db;
```

предоставляем все права на базу данных test_db пользователю dmitriy_s:

```
GRANT ALL PRIVILEGES ON DATABASE test_db to dmitriy_s;
```

выходим из оболочки PostgreSQL:

```
\q
```

```
postgres=# CREATE USER dmitriy_s WITH PASSWORD 'security_pass';
CREATE ROLE
postgres=# CREATE DATABASE test_db;
WARNING:  could not flush dirty data: Function not implemented
CREATE DATABASE
postgres=# GRANT ALL PRIVILEGES ON DATABASE test_db to dmitriy_s;
GRANT
postgres=# \q
postgres@DESKTOP-7BB5A8B:/home/dmitriy$
```

Проверяем подключение новым пользователем:

```
$ psql -Udmitriy_s test_db -h127.0.0.1
```

вводим для проверки простейшую команду:

```
SELECT 'olleH';
```

выходим из клиента psql:

```
\q
```

```
dmitriy@DESKTOP-7BB5A8B: ~
postgres@DESKTOP-7BB5A8B:/home/dmitriy$ psql -Udmitriy_s test_db -h127.0.0.1
Password for user dmitriy_s:
psql (12.11 (Ubuntu 12.11-0ubuntu0.20.04.1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
Type "help" for help.

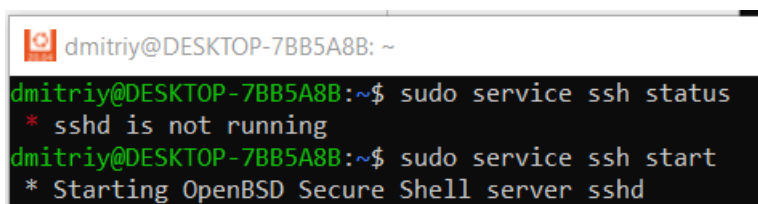
test_db=> SELECT 'olleH';
?column?
-----
 olleH
(1 row)

test_db=> \q
postgres@DESKTOP-7BB5A8B:/home/dmitriy$
```

Подключаться будем к серверу с базой данных по SSH. Так что нам понадобится демон sshd. Проверяем статус демона sshd, если он отключен запускаем его:

```
$ sudo service ssh status
```

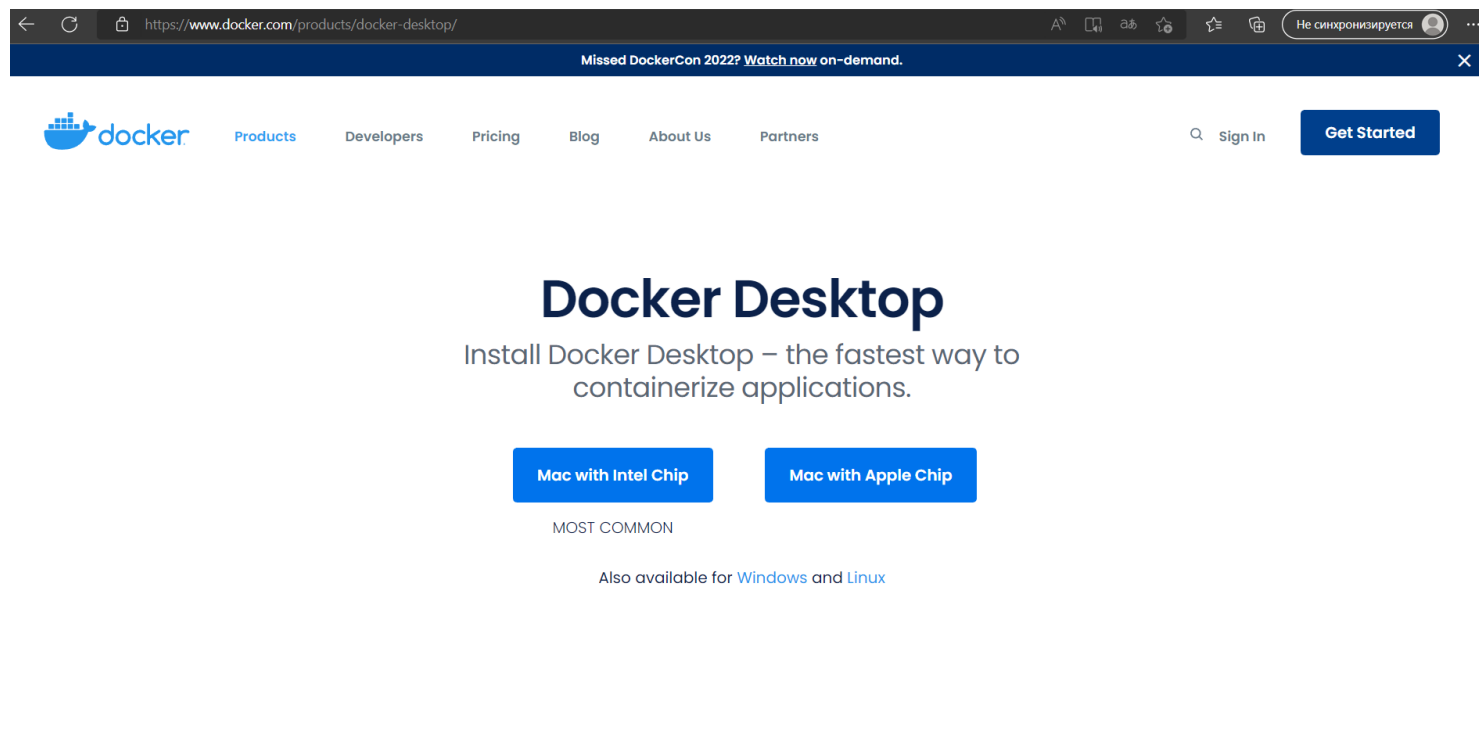
```
$ sudo service ssh start
```

A terminal window with a dark background and light green text. The window title bar shows a red icon and the text 'dmitriy@DESKTOP-7BB5A8B: ~'. The terminal content shows two commands and their outputs. The first command is 'sudo service ssh status' which outputs '* sshd is not running'. The second command is 'sudo service ssh start' which outputs '* Starting OpenBSD Secure Shell server sshd'.

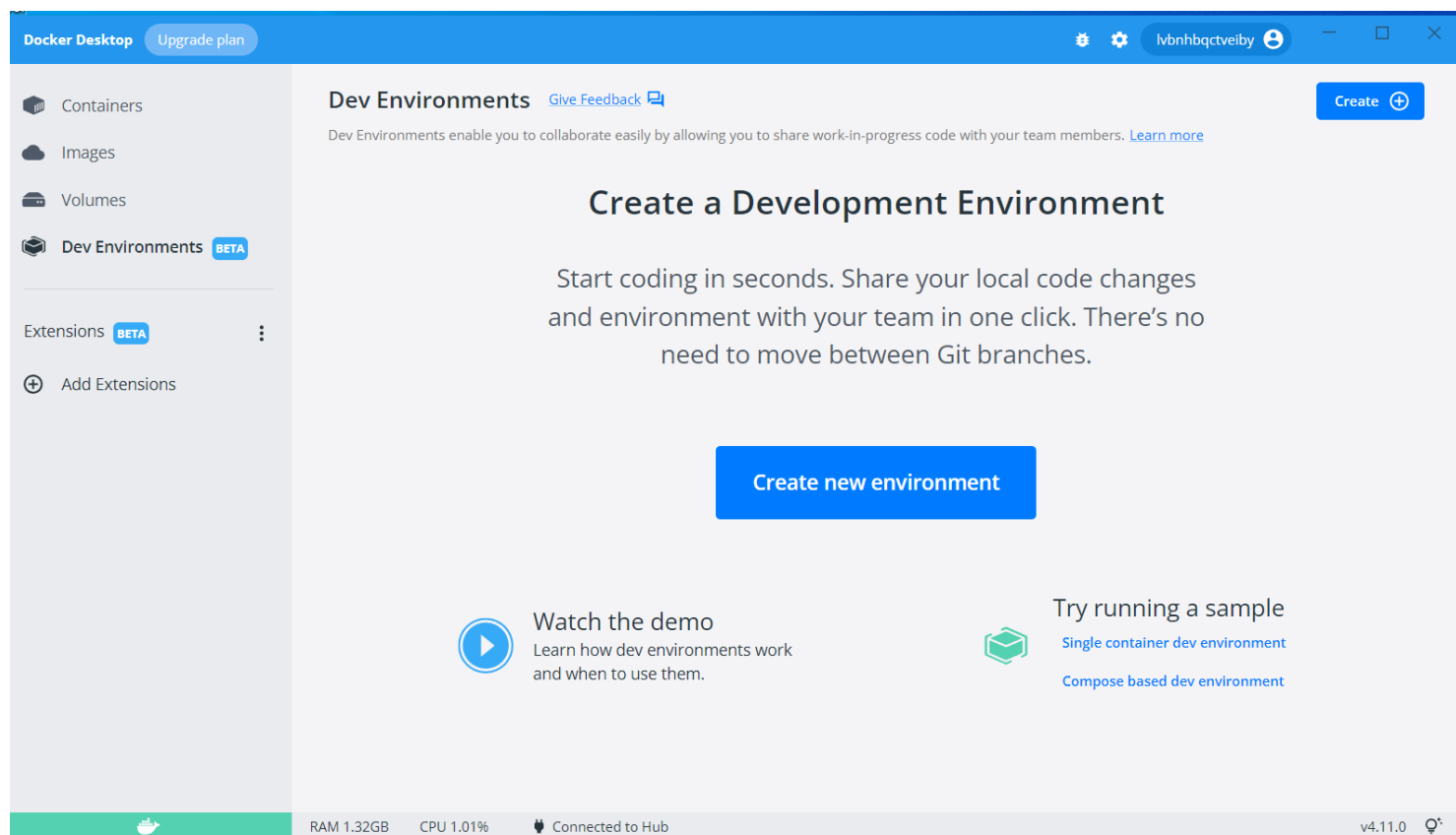
```
dmitriy@DESKTOP-7BB5A8B: ~  
dmitriy@DESKTOP-7BB5A8B:~$ sudo service ssh status  
* sshd is not running  
dmitriy@DESKTOP-7BB5A8B:~$ sudo service ssh start  
* Starting OpenBSD Secure Shell server sshd
```

После развертывания postgresql в WSL2 устанавливаем клиента pgadmin4 в Docker, но перед этим нужно скачать и установить Docker.

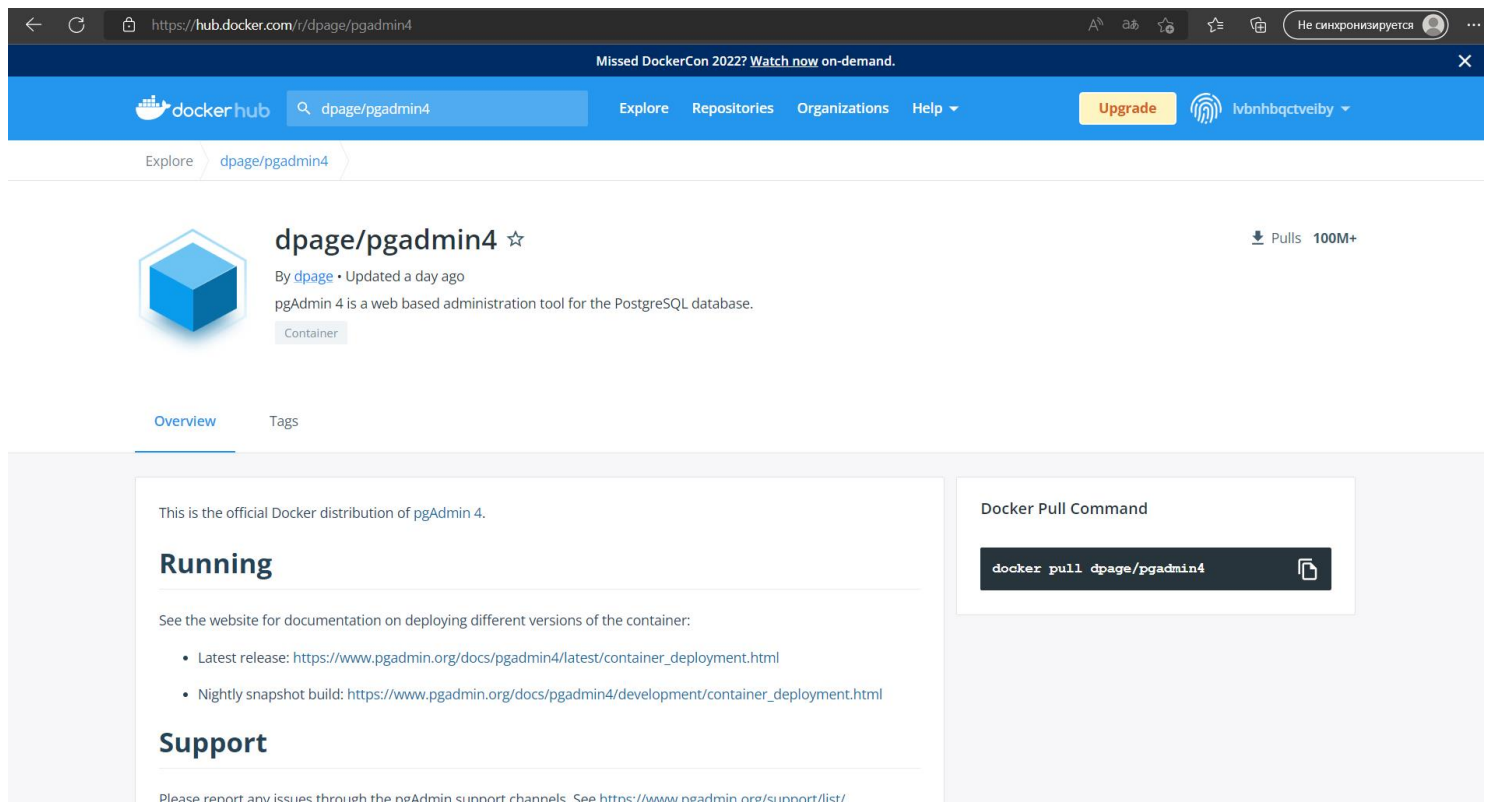
Переходим на сайт Docker, регистрируемся, скачиваем его и устанавливаем.



Далее запускаем Docker



Чтобы получить контейнер с pgadmin4, переходим в браузере на docker hub. Заходим в свою учетную запись и в поиске прописываем pgadmin4. В поисковом запросе выбираем dpage/pgadmin4.



https://hub.docker.com/r/dpage/pgadmin4

Missed DockerCon 2022? Watch now on-demand.

dockerhub

dpage/pgadmin4

Explore Repositories Organizations Help

Upgrade

lvnbhqbctveiby

Explore dpage/pgadmin4

dpage/pgadmin4 ☆

By dpage • Updated a day ago

pgAdmin 4 is a web based administration tool for the PostgreSQL database.

Container

Overview Tags

This is the official Docker distribution of pgAdmin 4.

Running

See the website for documentation on deploying different versions of the container:

- Latest release: https://www.pgadmin.org/docs/pgadmin4/latest/container_deployment.html
- Nightly snapshot build: https://www.pgadmin.org/docs/pgadmin4/development/container_deployment.html

Support

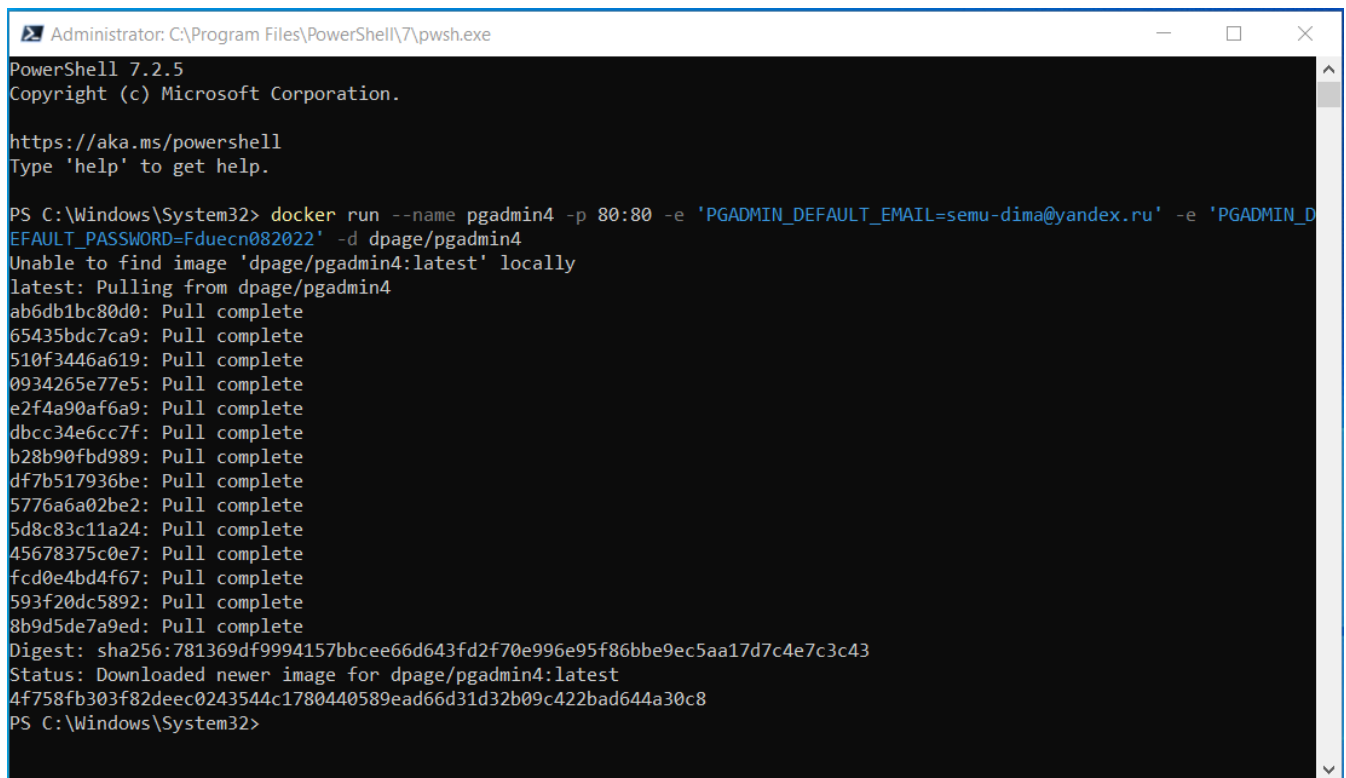
Please report any issues through the pgAdmin support channels. See <https://www.pgadmin.org/support/list/>

Docker Pull Command

```
docker pull dpage/pgadmin4
```

Для скачивания и запуска контейнера, открываем PowerShell и вводим следующую команду:

`docker run --name <имя контейнера> -p <порт> -e <переменная окружения: почта> -e <пароль> -d <название образа>`

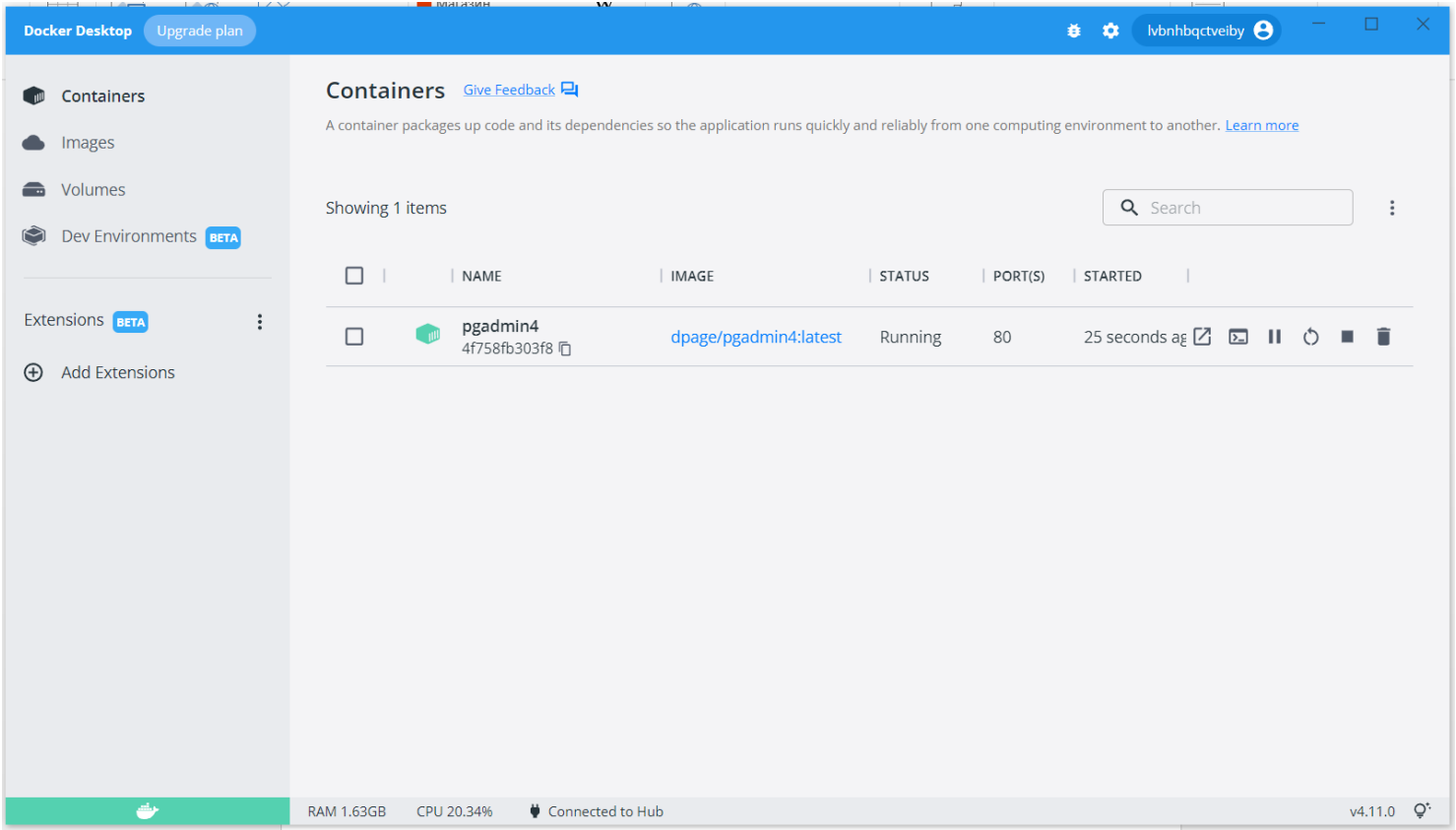


```
Administrator: C:\Program Files\PowerShell\7\pwsh.exe
PowerShell 7.2.5
Copyright (c) Microsoft Corporation.

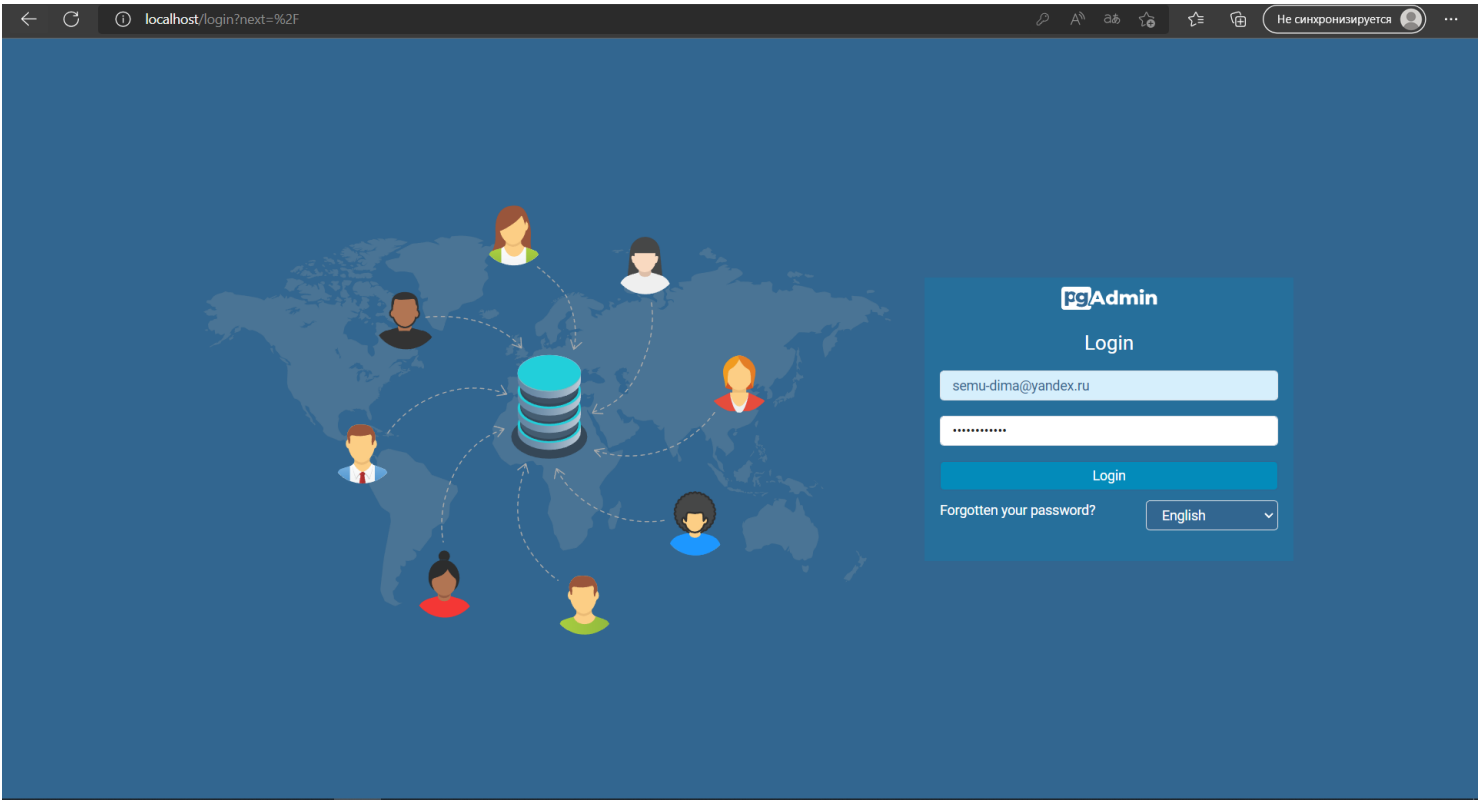
https://aka.ms/powershell
Type 'help' to get help.

PS C:\Windows\System32> docker run --name pgadmin4 -p 80:80 -e 'PGADMIN_DEFAULT_EMAIL=semu-dima@yandex.ru' -e 'PGADMIN_DEFAULT_PASSWORD=Fduecn082022' -d dpage/pgadmin4
Unable to find image 'dpage/pgadmin4:latest' locally
latest: Pulling from dpage/pgadmin4
ab6db1bc80d0: Pull complete
65435bdc7ca9: Pull complete
510f3446a619: Pull complete
0934265e77e5: Pull complete
e2f4a90af6a9: Pull complete
dbcc34e6cc7f: Pull complete
b28b90fbd989: Pull complete
df7b517936be: Pull complete
5776a6a02be2: Pull complete
5d8c83c11a24: Pull complete
45678375c0e7: Pull complete
fcd0e4bd4f67: Pull complete
593f20dc5892: Pull complete
8b9d5de7a9ed: Pull complete
Digest: sha256:781369df9994157bbcee66d643fd2f70e996e95f86bbe9ec5aa17d7c4e7c3c43
Status: Downloaded newer image for dpage/pgadmin4:latest
4f758fb303f82deec0243544c1780440589ead66d31d32b09c422bad644a30c8
PS C:\Windows\System32>
```


После скачивания и установки pgadmin4 в Docker мы увидим запущенный контейнер



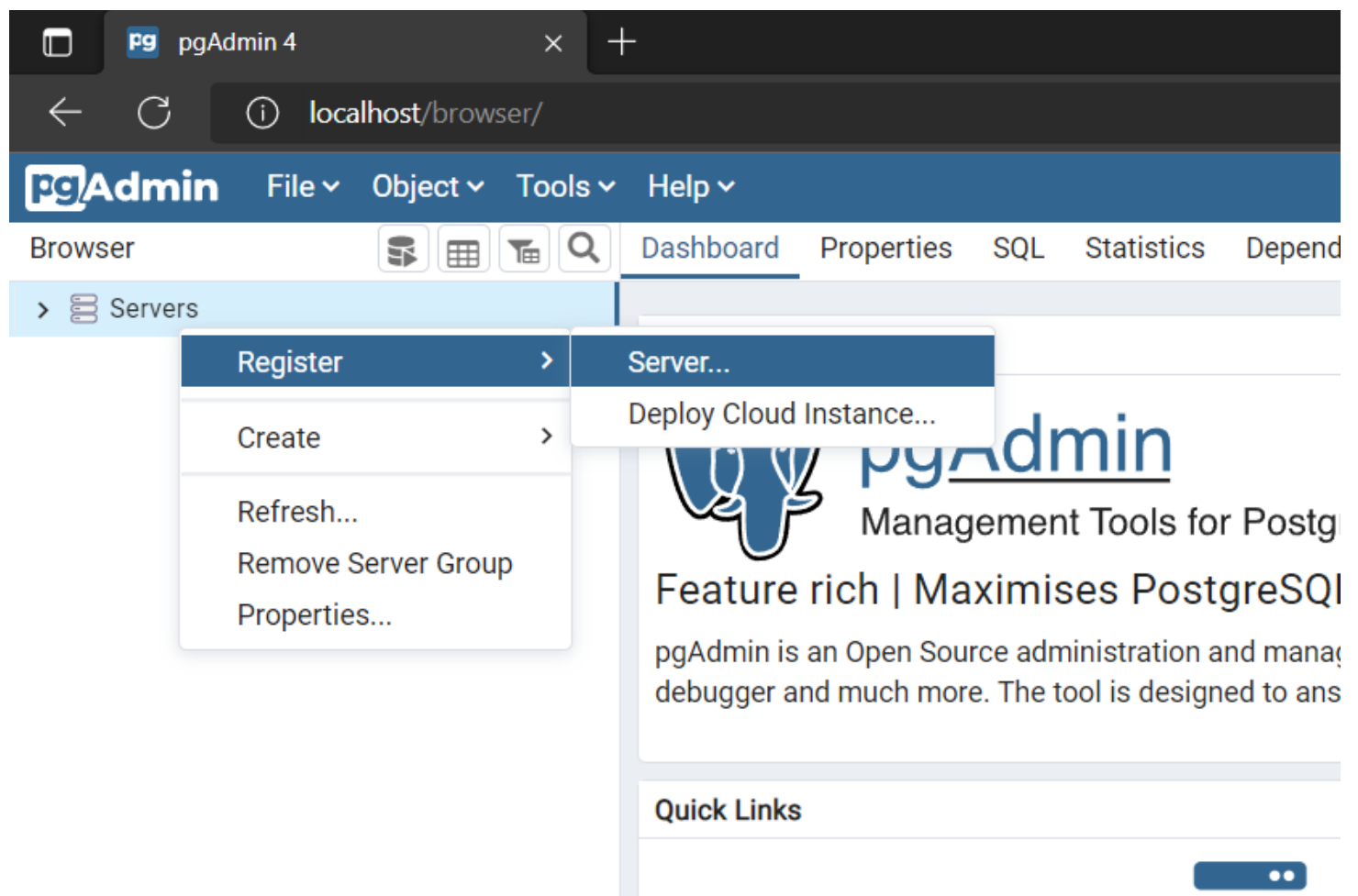
Далее в браузере перейдем по адресу с портом —р 80:80 <http://localhost:80>. В открывшемся веб-интерфейсе вводим почту и пароль, что задали в -е переменных окружения при выполнении команды в PowerShell docker run ...



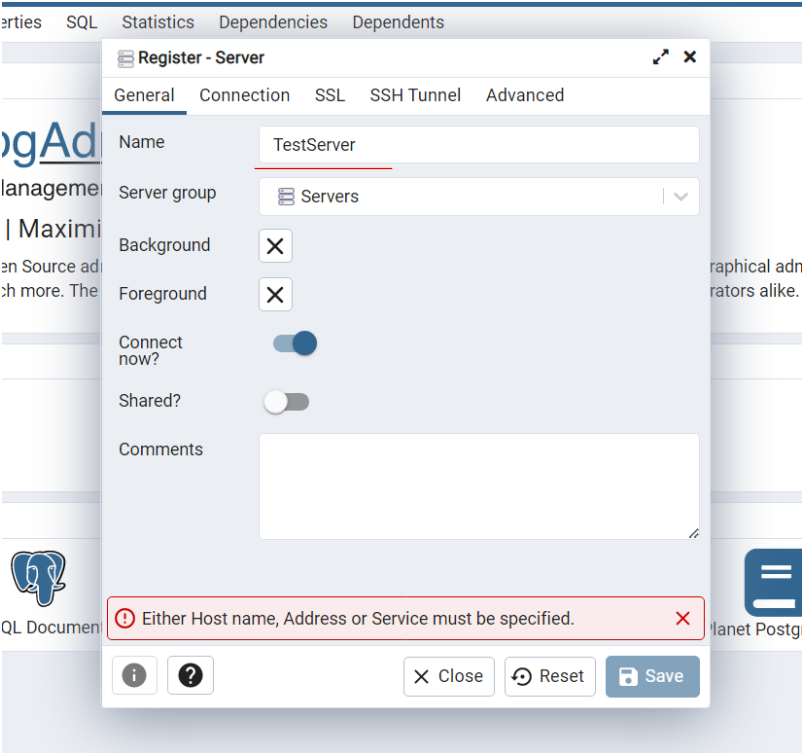
После успешного входа, попадаем на следующую страницу



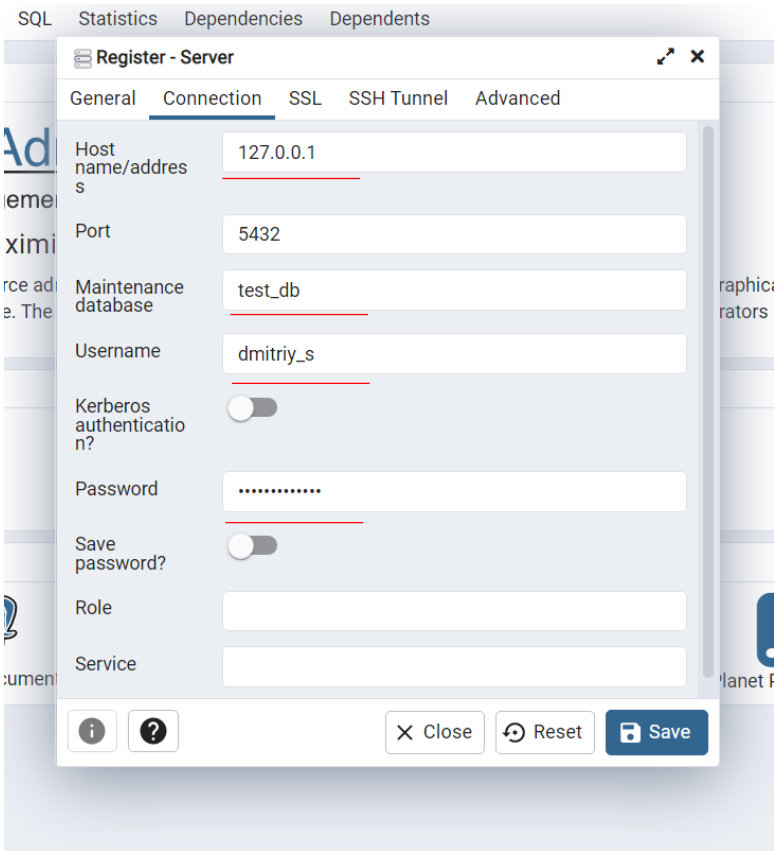
Далее регистрируем новый сервер. Правой кнопкой мыши по Servers -> Register -> Server...



В появившемся всплывающем окне на вкладке «Основные» прописываем имя сервера



На вкладке «Подключение» вводим имя хоста/адрес, имя созданной базы данных, имя пользователя базы данных и пароль от базы данных



Далее переходим на вкладку «Туннель SSH», переключаем ползунок, вводим хост туннеля*, вводим имя пользователя Ubuntu и пароль

Register - Server

General Connection SSL **SSH Tunnel** Advanced

Use SSH tunneling ☒

Tunnel host 192.168.100.186

Tunnel port 22

Username dmitriy

Authentication ☒ Password ☐ Identity file

Identity file

Password

Save password? ☐

Close Reset Save

*Для того, чтобы узнать хост туннеля в WSL2 вводим команду:

\$ ip a

```
dmitriy@DESKTOP-7BB5A8B: ~  
dmitriy@DESKTOP-7BB5A8B:~$ ip a  
3: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 group default qlen 1  
    link/ether 98:29:a6:3e:36:43  
    inet 192.168.100.186/22 brd 192.168.103.255 scope global dynamic  
        valid_lft 691189sec preferred_lft 691189sec  
    inet6 fe80::9c56:9ff3:b0fe:5b4e/64 scope link dynamic  
        valid_lft forever preferred_lft forever  
18: eth1: <> mtu 1400 group default qlen 1  
    link/ether 44:45:53:54:4f:53  
    inet 169.254.8.145/16 brd 169.254.255.255 scope global dynamic  
        valid_lft forever preferred_lft forever  
    inet6 fe80::256f:7574:7644:891/64 scope link dynamic  
        valid_lft forever preferred_lft forever  
52: eth2: <BROADCAST,MULTICAST,UP> mtu 1500 group default qlen 1  
    link/ether 00:15:5d:ab:95:fa  
    inet 169.254.84.211/16 brd 169.254.255.255 scope global dynamic  
        valid_lft forever preferred_lft forever  
    inet6 fe80::681d:733b:e5d9:54d3/64 scope link dynamic  
        valid_lft forever preferred_lft forever  
56: eth3: <BROADCAST,MULTICAST,UP> mtu 1500 group default qlen 1  
    link/ether 0a:00:27:00:00:38  
    inet 192.168.56.1/24 brd 192.168.56.255 scope global dynamic  
        valid_lft forever preferred_lft forever  
    inet6 fe80::39c7:2aa:62db:a0fa/64 scope link dynamic  
        valid_lft forever preferred_lft forever  
1: lo: <LOOPBACK,UP> mtu 1500 group default qlen 1  
    link/loopback 00:00:00:00:00:00  
    inet 127.0.0.1/8 brd 127.255.255.255 scope global dynamic  
        valid_lft forever preferred_lft forever
```

После нажатия на кнопку «Сохранить», ждем подключения к серверу. После успешного подключения слева в проводнике появляется наш сервер и база данных.

