

Настройка SSL сертификатов

OpenSSH

(1) Enable SSH connection via db.Dockerfile

составляем докерфайл, чтобы при загрузке образа запускалась возможность создавать ssh соединение напрямую с контейнером

```
db.Dockerfile > ...
1 FROM postgres
2
3 ARG ssh_user
4 ARG ssh_password
5
6 RUN useradd -ms /bin/bash $ssh_user
7 RUN echo "$ssh_user:$ssh_password" | chpasswd
8
9 RUN apt update && apt -y install openssh-server
10 RUN mkdir -p /var/run/sshd
11 RUN sed -i 's/#PermitRootLogin prohibit-password/PermitRootLogin yes/' /etc/ssh/sshd_config
12 CMD ["/usr/sbin/sshd", "-D"]
```

(2) Open port in docker compose

открываем соответствующий порт в контейнере, остальные порты автоматически недоступны по политикам доступа Docker
создаем специального пользователя для защищенной работы

```
12 db:
13   build:
14     context: .
15     dockerfile: db.Dockerfile
16     args:
17       ssh_user: guest
18       ssh_password: ${ROOT_PASS}
19   ports:
20     - "5433:5432"
21     - "2288:22"
22   environment:
23     POSTGRES_PASSWORD: ${POSTGRES_PASSWORD}
24     POSTGRES_USER: ${POSTGRES_USER}
25     POSTGRES_DB: ${POSTGRES_DB}
26     PGDATA: ${POSTGRES_PGDATA}
27   volumes:
28     - ./db/data:/var/lib/postgresql/data
29     - ./db/initdb.d:/docker-entrypoint-initdb.d
30   networks:
31     localnet:
32       ipv4_address: ${DB_ADDRESS}
```

(3) Check ssh connection

с клиентского окна подключаемся к сети контейнера через ssh

```
guest@ddb7b8757981: ~
$ ssh -p 2288 guest@localhost
The authenticity of host '[localhost]:2288 ([127.0.0.1]:2288)' can't be established.
ED25519 key fingerprint is SHA256:yp9w5KYd/Pi+IoJxNx4SQZ3SGplVKTvlokbiYSUxpOY.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[localhost]:2288' (ED25519) to the list of known hosts.
guest@localhost's password:
Linux ddb7b8757981 5.15.146.1-microsoft-standard-WSL2 #1 SMP Thu Jan 11 04:09:03 UTC 2024 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
guest@ddb7b8757981:~$ ls
guest@ddb7b8757981:~$ ls /
bin  dev  etc  lib  media  opt  root  sbin  sys  usr
boot  docker-entrypoint-initdb.d  home  lib64  mnt  proc  run  srv  tmp  var
guest@ddb7b8757981:~$
```

Настройка SSL

(4) Создаем pg_hba.conf

```
102 #
103 # If you want to allow non-local connections, you need to add more
104 # "host" records.  In that case you will also need to make PostgreSQL
105 # listen on a non-local interface via the listen_addresses
106 # configuration parameter, or via the -i or -h command line switches.
107
108 # CAUTION: Configuring the system for local "trust" authentication
109 # allows any local user to connect as any PostgreSQL user, including
110 # the database superuser.  If you do not trust all your local users,
111 # use another authentication method.
112
113
114 # TYPE      DATABASE      USER      ADDRESS      METHOD
115
116 local      all             all                                     trust
117 hostssl    all             all        all           cert
118 host       all             all        all           reject
119
120 # Allow replication connections from localhost, by a user with the
121 # replication privilege.
122 local      replication    all                                     trust
123 host       replication    all        127.0.0.1/32  trust
124 host       replication    all        ::1/128       trust
125
126 # host all all all scram-sha-256
```

(5) Создаем sh скрипт для генерации ключей сервера и клента

генерация ключей сервера

```

openssl > postgres > $ generate_server_keys.sh
1  #!/bin/bash
2
3  KEY_SECRET=2123wdqwid2e98qdh3iud3
4  openssl req -new -text -passout pass:$KEY_SECRET -subj /CN=localhost -keyout privkey.pem -out server.req
5  openssl rsa -in privkey.pem -passin pass:$KEY_SECRET -out server.key
6  openssl req -x509 -in server.req -text -key server.key -out server.crt

```

генерация ключей клиента

```

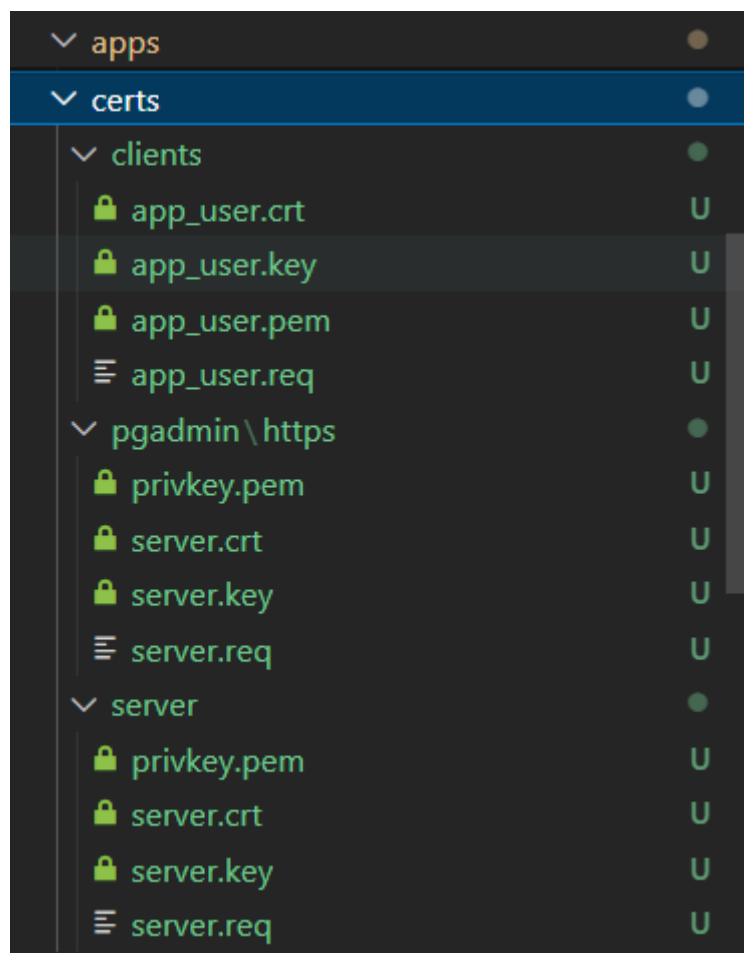
openssl > postgres > $ generate_appuser_keys.sh
1  #!/bin/bash
2
3  KEY_SECRET=12345678
4  USERNAME=app_user
5  openssl req -new -text -passout pass:$KEY_SECRET -subj /CN=app_user -keyout $USERNAME.pem -out $USERNAME.req
6  openssl rsa -in $USERNAME.pem -passin pass:$KEY_SECRET -out $USERNAME.key
7  openssl req -x509 -in $USERNAME.req -text -key $USERNAME.key -out $USERNAME.crt

```

[Важно]

для генерации https соединения можно воспользоваться файлом для сервера

(6) Генерируем ключи



(7) Заполняем конфиги docker-compose

(7.1) для сервера

```

12 db:
13   build:
14     context: .
15     dockerfile: db.Dockerfile
16     args:
17       ssh_user: guest
18       ssh_password: ${ROOT_PASS}
19   ports:
20     - "5433:5432"
21     - "2288:22"
22   environment:
23     POSTGRES_PASSWORD: ${POSTGRES_PASSWORD}
24     POSTGRES_USER: ${POSTGRES_USER}
25     POSTGRES_DB: ${POSTGRES_DB}
26     PGDATA: ${POSTGRES_PGDATA}
27   volumes:
28     - ./db/data:/var/lib/postgresql/data
29     - ./db/initdb.d:/docker-entrypoint-initdb.d
30     - ./config/pg_hba.conf:/var/lib/postgresql/data/pg_hba.conf:ro
31   networks:
32     localnet:
33       ipv4_address: ${DB_ADDRESS}

```

(7.2) для админки

```

34 db-ui:
35   image: dpage/pgadmin4
36   ports:
37     - "5050:80"
38     - "5053:443"
39   environment:
40     PGADMIN_DEFAULT_EMAIL: ${PGADMIN_EMAIL}
41     PGADMIN_DEFAULT_PASSWORD: ${PGADMIN_PWD}
42     PGADMIN_LISTEN_ADDRESS: '0.0.0.0'
43     PGADMIN_ENABLE_TLS: true
44     PGADMIN_CONFIG_SERVER_MODE: "False"
45     PGADMIN_CONFIG_MASTER_PASSWORD_REQUIRED: "False"
46   volumes:
47     - ./db/pgadmin:/var/lib/pgadmin
48     - ./config/pgadmin/servers.json:/pgadmin4/servers.json
49     # HTTPS certificates mapping. NB! server.crt -> server.cert, server.key -> server.key
50     - ./certs/pgadmin/https/server.crt:/certs/server.cert:ro
51     - ./certs/pgadmin/https/server.key:/certs/server.key:ro
52     # TLS superuser client certificates mapping
53     - ./certs/clients/app_user.crt:/certs/app_user.crt:ro
54     - ./certs/clients/app_user.key:/certs/app_user.key:ro
55   restart: unless-stopped
56   deploy:
57     resources:
58       limits:
59         cpus: '0.5'
60         memory: 1G
61   networks:
62     localnet:
63       ipv4_address: ${ADMIN_ADDRESS}
64

```

(8) Проверка ssl на сервере

```

$ docker exec -it 2c8 psql -U app_user "sslmode=require dbname=wirehouse_db"
psql (16.2 (Debian 16.2-1.pgdg120+2))
Type "help" for help.

wirehouse_db=# _

```

(9) Админ панель с самоподписным сертификатом

