**1. Асинхронная репликация**

1. Запустить master из dockre-compose

postgres:

container\_name: postgres

image: postgres:15.4-alpine3.18

environment:

POSTGRES\_DB: "otusdb"

POSTGRES\_USER: "sa"

POSTGRES\_PASSWORD: "medtex"

volumes:

- ../migrations:/docker-entrypoint-initdb.d

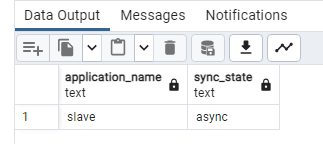
- otusdb-data:/var/lib/postgresql/data

ports:

- "5432:5432"

3. Запустить скрипт

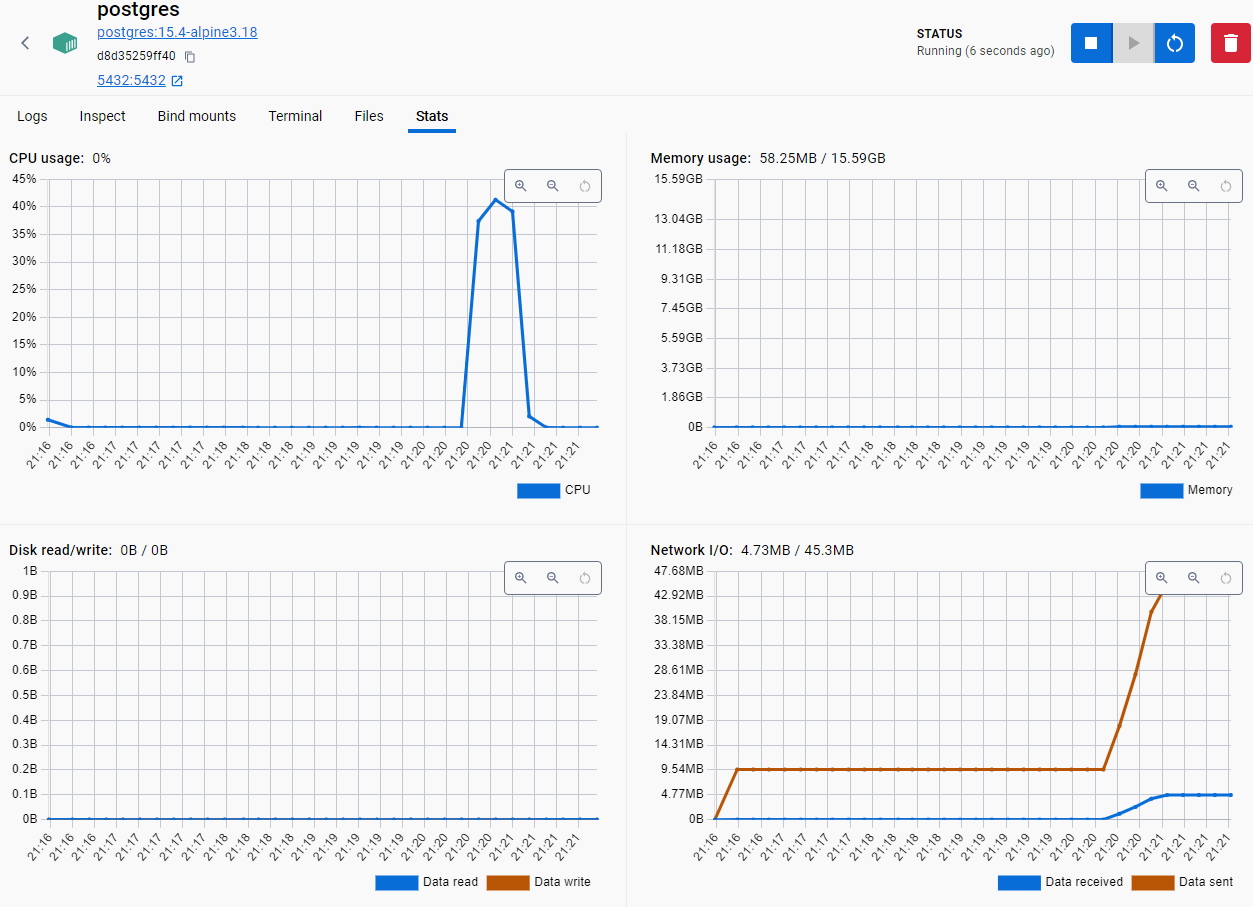
bash -c "  
 echo \"host replication replicator 172.18.0.0/16 md5\" >> /var/lib/postgresql/data/pg\_hba.conf &&  
 echo \"wal\_level = replica\" >> /var/lib/postgresql/data/postgresql.conf &&  
 echo \"max\_wal\_senders = 4\" >> /var/lib/postgresql/data/postgresql.conf &&  
 echo \"ssl = off\" >> /var/lib/postgresql/data/postgresql.conf  
 "  
  
4. Перезапустить master  
  
docker restart postgres  
  
5. Сделать backup для реплик  
  
docker exec -it postgres bash  
  
mkdir /pgslave  
  
pg\_basebackup -h postgres -D /pgslave -U replicator -v -P --wal-method=stream  
  
6. Скопировать dir на хост  
  
docker cp postgres:/pgslave pgslave  
  
7. Скопировать standby.signal, чтобы сделать реплику  
  
#F5 to ($PWD)\pgslave  
  
8. Изменить postgresql.conf на реплике  
  
primary\_conninfo = 'host=postgres port=5432 user=replicator password=medtex application\_name=slave'  
  
9. Запустить реплику  
docker run -dit -v $PWD/pgslave/:/var/lib/postgresql/data -e POSTGRES\_PASSWORD=medtex -p 15433:5432 --network=docker\_default --restart=unless-stopped --name=pgslave postgres:15.4-alpine3.18  
  
8 Проверить мастер  
select application\_name, sync\_state from pg\_stat\_replication;



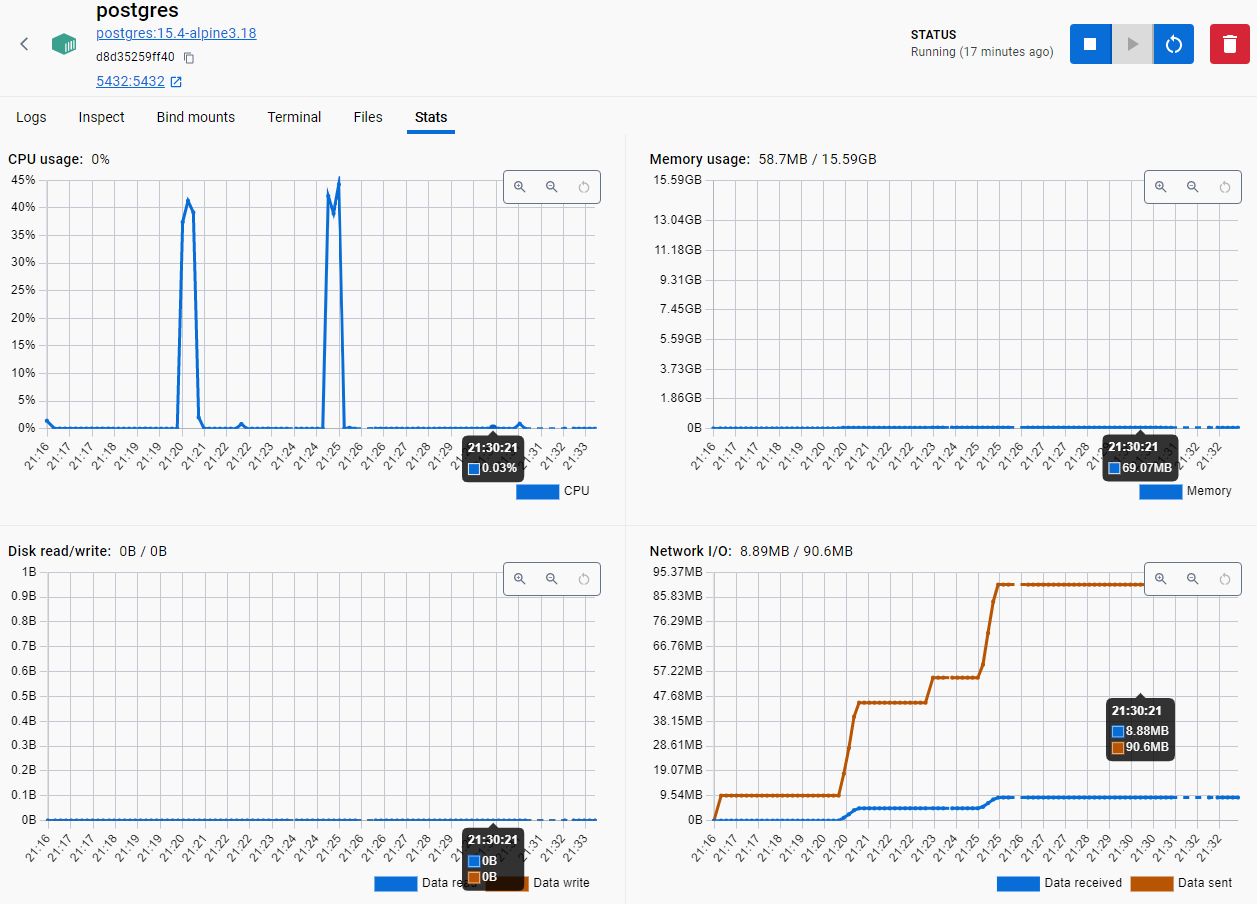
Проверяем передачу нагрузки с мастера на реплику

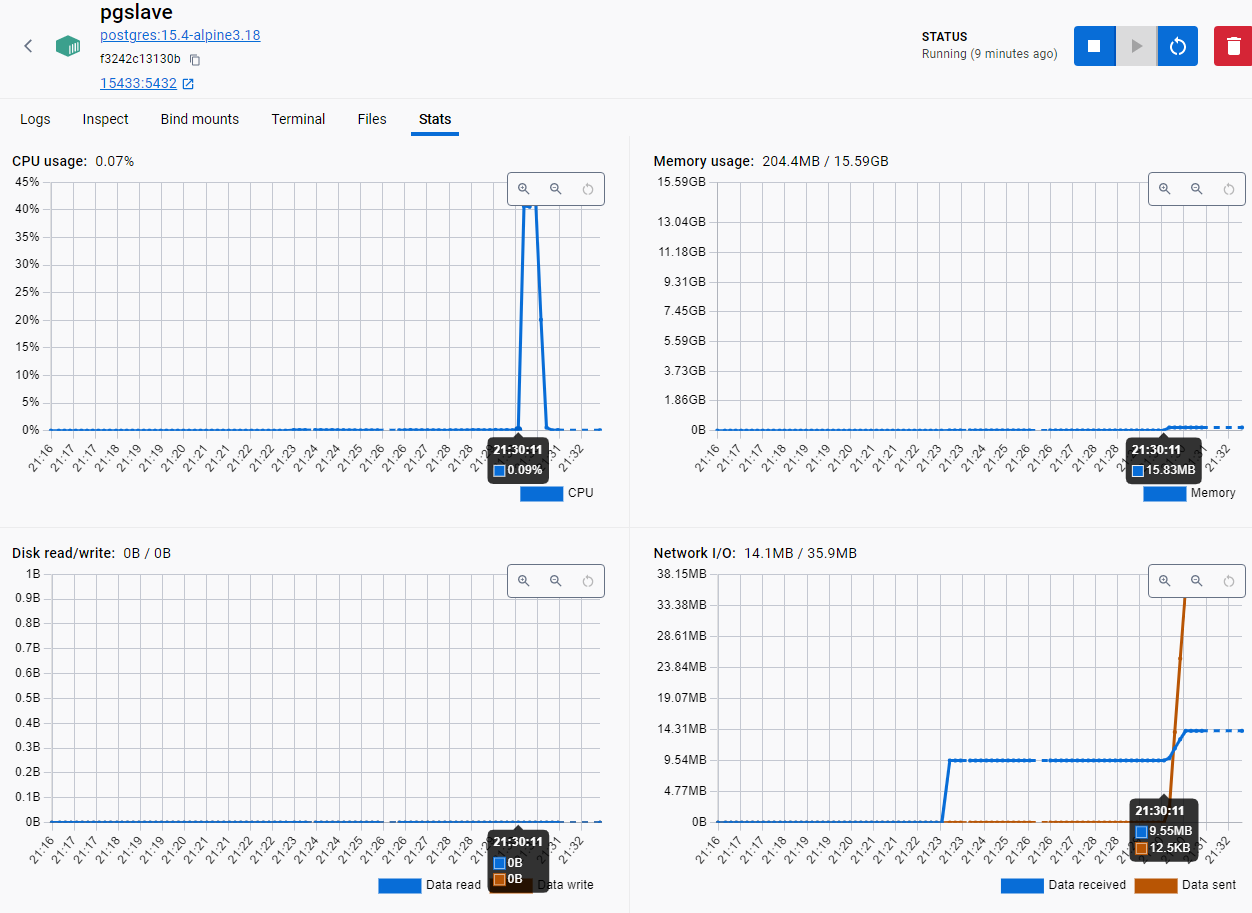
.**/ab -n 10000 -c 100 http://localhost:5266/user/search?Name=%25%D0%B5%D0%B2"&"Surname=%25%D1%80%D0%B0%D0%BC%D0%BE%25**

**Нагрузка на мастере до включения реплики**



**Нагрузка на мастере после включения реплики**

****

Нагрузка перешла на реплику

**2. Синхронная репликация**

---

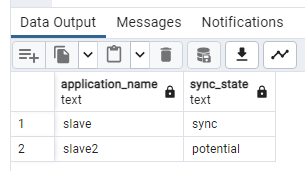
1. Запустить 2ую реплику  
docker exec -it postgres bash  
  
mkdir /pgslave2  
  
pg\_basebackup -h postgres -D /pgslave2 -U replicator -v -P --wal-method=stream  
  
docker cp postgres:/pgslave2 pgslave2  
  
#Copy standby.signal, to make replica  
  
primary\_conninfo = 'host=postgres port=5432 user=replicator password=medtex application\_name=slave2'  
  
docker run -dit -v $PWD/pgslave2/:/var/lib/postgresql/data -e POSTGRES\_PASSWORD=medtex -p 25433:5432 --network=docker\_default --restart=unless-stopped --name=pgslave2 postgres:15.4-alpine3.18

2. Включаем синхронную репликацию на мастере:

Изменить postgresql.conf

synchronous\_commit = on  
synchronous\_standby\_names = 'FIRST 1 (slave, slave2)'

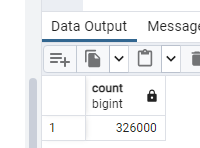
Перезапустить конфигурацию  
select pg\_reload\_conf();



3. Создаем нагрузку и останавливаем мастер

4. Смотрим результат транзакции на обоих репликах

SELECT count(\*) FROM public.users;



5. Смотрим результат транзакции на мастере

SELECT count(\*) FROM public.users;

Результат аналогичен репликам. **Потерь транзакций нет.**

**3. Повышение реплики**

1. Запромоутим реплику pgslave  
  
select \* from pg\_promote();

Изменить postgresql.conf

synchronous\_commit = on  
synchronous\_standby\_names = 'ANY 1 (postgres, slave2)'  
  
2. Подключим вторую реплику к новому мастеру  
  
primary\_conninfo = 'host=pgslave port=5432 user=replicator password=pass application\_name=slave2'

3. Проверяем что реплика переключилась на новый мастер

select application\_name, sync\_state from pg\_stat\_replication;

