

Повторяем гайд до 11 пункта включительно <https://github.com/OtusTeam/highload/blob/master/lessons/02/05/live/guide.md>

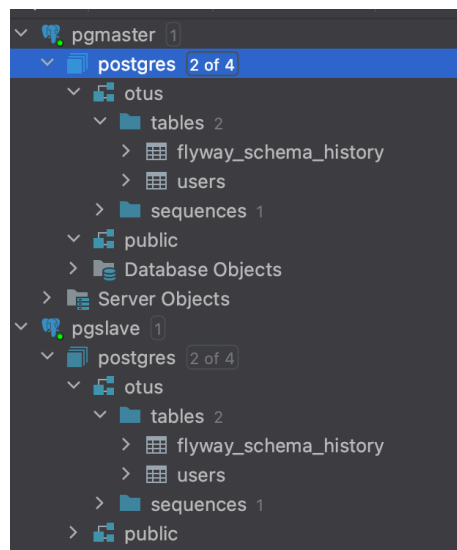
Создаем дамп базы из прошлого дз:

```
pg_dump --file=/Users/ydpolvt/postgres_localhost-2024_07_14_18_01_56-dump.sql --username=postgres --host=localhost --port=15432
```

Грузим дамп на pgmaster:

```
psql --dbname=postgres --username=postgres --host=localhost --port=5432 -d postgres -f /Users/ydpolvt/postgres_localhost-2024_07_14_18_01_56-dump.sql
```

В итоге получится одинаковый набор данных на обоих инстансах базы:



С помощью Jmeter нагрузочного тестирования и прошлого задания проводим тест программы только с pgmaster, затем переключаем нагрузку на чтение на pgslave и проводим тест снова.

До:

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
ae04c7371e74	otus_highload_3-backend-1	104.86%	1.293GiB / 7.657GiB	16.89%	743MB / 541MB	0B / 81.9kB	137
3a9835510922	pgslave	0.36%	154MiB / 7.657GiB	1.96%	248MB / 14.1MB	0B / 4.1kB	6
712c9931bfda	pgmaster	393.97%	378MiB / 7.657GiB	4.82%	616MB / 965MB	0B / 81.1MB	107
9cd80e93d0b8	otus_highload_2-service-db-1	0.00%	55.97MiB / 7.657GiB	0.71%	4.04MB / 381MB	230MB / 18MB	8

После:

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
3fd588f63878	otus_highload_3-backend-1	104.14%	1.2GiB / 7.657GiB	15.67%	335MB / 241MB	0B / 32.8kB	137
3a9835510922	pgslave	270.31%	353.6MiB / 7.657GiB	4.51%	254MB / 346MB	0B / 4.1kB	105
712c9931bfda	pgmaster	0.00%	166.4MiB / 7.657GiB	2.12%	627MB / 1.34GB	0B / 81.1MB	17
9cd80e93d0b8	otus_highload_2-service-db-1	0.00%	55.96MiB / 7.657GiB	0.71%	4.04MB / 381MB	230MB / 18.1MB	8

Как видно, нагрузка перешла на другой инстанс.

Повторяем 7-11 пункты гайда, чтобы сделать еще одну асинхронную реплику pgslave2.

Далее вносим изменения в pgmaster/postgresql.conf :

```
synchronous_commit = on  
synchronous_standby_names = 'ANY 1 (pgslave, pgslave2)'
```

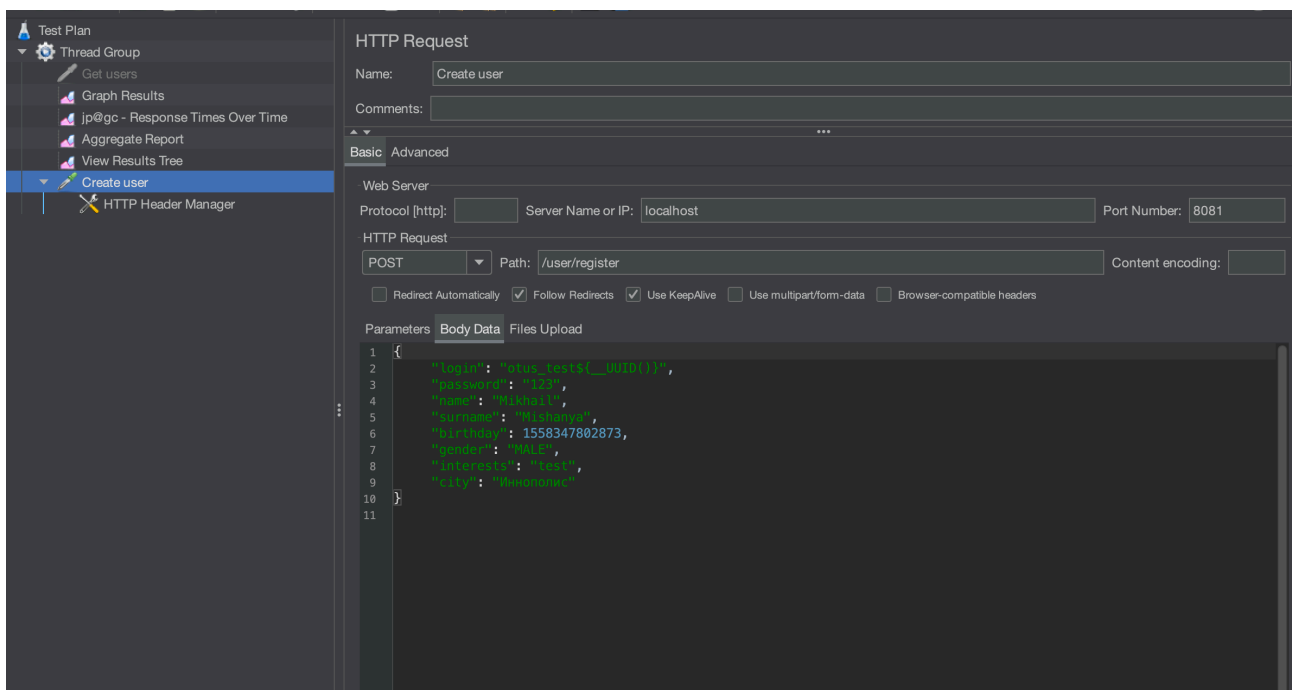
Перечитываем конфиг:

```
docker exec -it pgmaster su - postgres -c  
psql select pg_reload_conf();  
exit;
```

По итогу получаем:

```
postgres=# select application_name, sync_state from pg_stat_replication;  
 application_name | sync_state  
-----+-----  
 pgslave          | quorum  
 pgslave2         | quorum  
(2 rows)
```

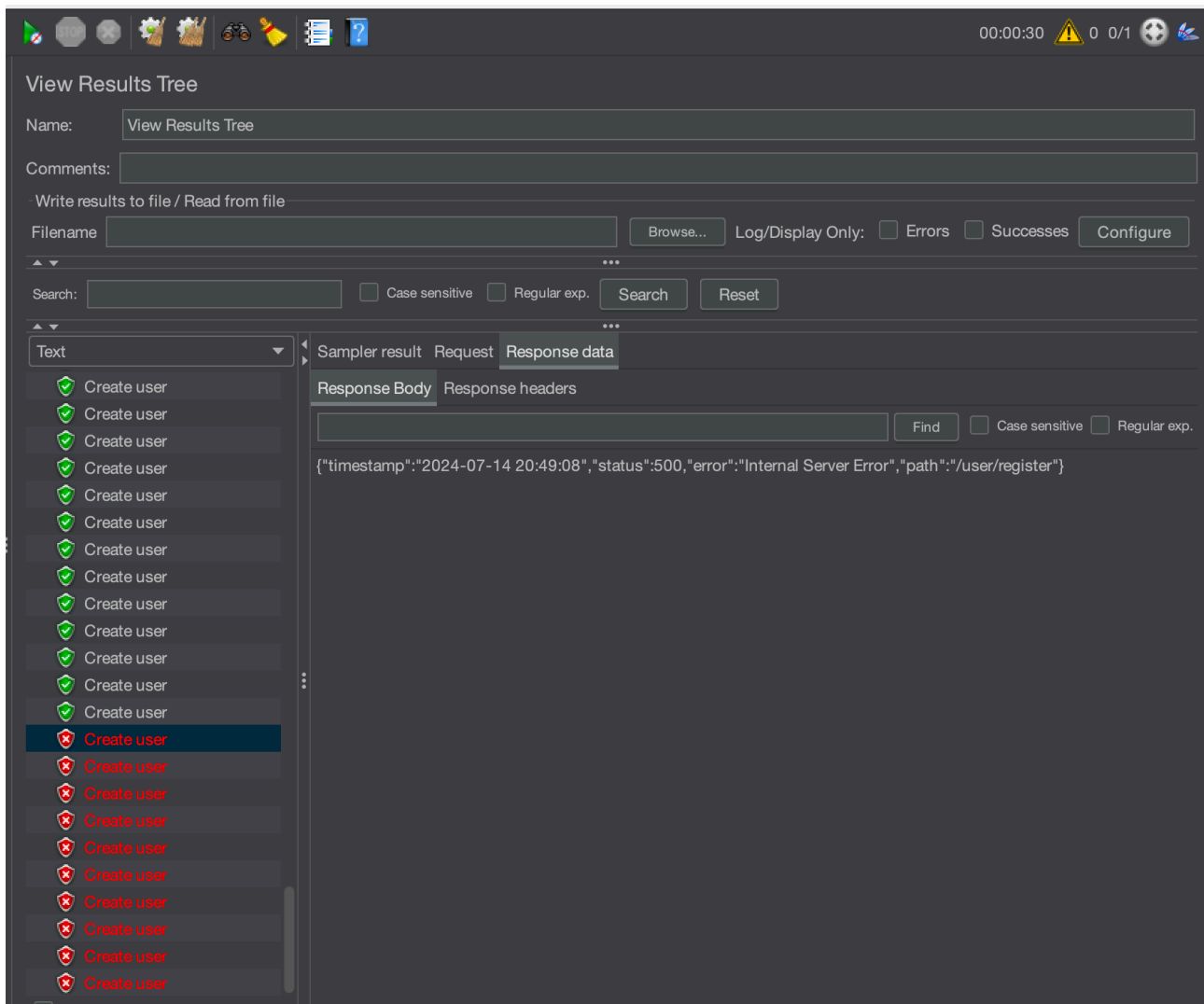
После этого создаем загрузку на запись с помощью Jmeter:



После этого останавливаем pgmaster. Как видно что после остановки pgmaster посыпались ошибки о невозможности создания нового пользователя:

Aggregate Report
Name: Aggregate Report
Comments:
Write results to file / Read from file
Filename Browse... Log/Display Only: Errors Successes Configure

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Through...	Received...	Sent KB/...
Create u...	128	240	225	252	267	708	3	3018	7,81%	4,2/sec	1,87	1,76
TOTAL	128	240	225	252	267	708	3	3018	7,81%	4,2/sec	1,87	1,76



После этого промодулим pgslave2 до мастера:

```
[0485NBT0VW9W70Q:pgmaster ydpolivt$ docker exec -it pgcslave2 su - postgres -c psql
psql (15.3 (Debian 15.3-1.pgdg110+1))
Type "help" for help.

postgres=# select pg_promote();
pg_promote
-----
 t
(1 row)

postgres=# exit;
```

Далее настраиваем репликацию на pgslave2. Изменяем конфиг и перечитываем его:
synchronous_commit = on
synchronous_standby_names = 'ANY 1 (pgslave)'

```
docker exec -it pgcslave2 su - postgres -c psql
select pg_reload_conf();
exit;
```

Подключаем реплику pgslave. Изменяем конфиг и перечитываем его:

```
primary_conninfo = 'host=pgcslave2 port=5432 user=replicator password=pass
application_name=pgslave'
```

```
docker exec -it pgslave su - postgres -c psql
select pg_reload_conf();
```

Теперь у нас настроена репликация:

```
[0485NBT0VW9W70Q:pgslave ydpolivt$ docker exec -it pgcslave2 su - postgres -c psql
psql (15.3 (Debian 15.3-1.pgdg110+1))
Type "help" for help.

postgres=# select application_name, sync_state from pg_stat_replication;
 application_name | sync_state 
-----+-----
 pgslave          | quorum
(1 row)
```

После этого перезапускаю докер контейнер с основным приложением, заменив креды pgmaster на pgslave2.

После этого даем ту же самую нагрузку и как результат запись снова работает:

Aggregate Report

Name: Aggregate Report

Comments:

- Write results to file / Read from file

Filename Browse... Log/Display Only:
 ☐ Errors
 ☒ Successes
 Configure

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Through...	Received...	Sent KB/...
Create u...	180	173	166	186	202	251	158	506	0,00%	5,8/sec	2,47	2,46
TOTAL	180	173	166	186	202	251	158	506	0,00%	5,8/sec	2,47	2,46

View Results Tree

Name: [View Results Tree](#)

Comments:

- Write results to file / Read from file

Filename

Log/Display Only:

Successes

Configure

Search:

☐ Case sensitive☐ Regular exp.

Search

Reset

Text

Sampler result

Request

Response data

Response Body

Response headers

☐ Case sensitive☐ Regular exp.