#### 1. Запустить контейнеры

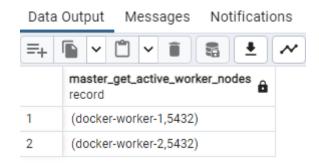
docker-compose -f ./shards-compose.yml up --scale worker=2 -d

## 2. Создадим распределенную (шардированную) таблицу:

SELECT create\_distributed\_table('dialogs', 'author\_id');

### 3. Посмотрим, видит ли координатор шарды:

SELECT shard\_count FROM citus\_tables WHERE table\_name::text = 'dialogs'; SELECT master\_get\_active\_worker\_nodes();

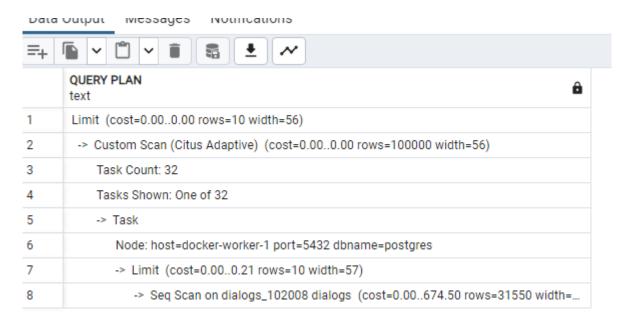


### 4. Наполним данными

```
insert into dialogs(author_id, user_id, message)
select
   i,
        i+1,
   md5(random()::text)
from generate series(1, 1000000) as i;
```

#### 5. Посмотрим план запроса

explain select \* from dialogs limit 10;

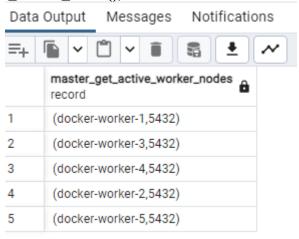


#### 6. Добавить шардов

docker-compose -f ./shards-compose.yml up --scale worker=5 -d

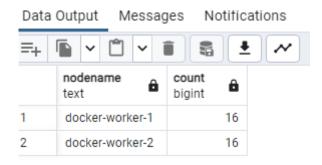
#### 7. Посмотрим, видит ли координатор новые шарды:

SELECT master\_get\_active\_worker\_nodes();



## 8. Проверим, на каких узлах лежат сейчас данные:

SELECT nodename, count(\*)
FROM citus\_shards GROUP BY nodename;



## 9. Установим wal\_level = logical чтобы узлы могли переносить данные:

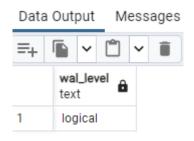
alter system set wal\_level = logical; SELECT run\_command\_on\_workers('alter system set wal\_level = logical');

### 10. Перезапускаем все узлы в кластере, чтобы применить изменения wal\_level.

B pg\_hba.conf шардов установить host all all trust docker-compose -f ./shards-compose.yml restart

### 11. Проверим, что wal\_level изменился:

show wal\_level;



# 12. В таблице нет первичного ключа, п.э. устанавливаем режим шардирования

```
SELECT rebalance_table_shards(
  'dialogs',
    shard_transfer_mode => 'force_logical'
);
```

## 13. Запустим ребалансировку:

SELECT citus\_rebalance\_start();

### 14. Проверяем, что данные равномерно распределились по шардам:

SELECT nodename, count(\*)
FROM citus\_shards GROUP BY nodename;

