

Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский университет
ИТМО»

Факультет Информационных технологий и программирования

Лабораторная работа №3
Администрирование и оптимизация

Выполнил: студент группы
М3203

Костыгов Андрей Константинович

Проверила:

Шевчик Софья Владимировна

Санкт-Петербург
2024 г.

Задачи: повышение производительности и отказоустойчивости СУБД.

Запросы

Сначала напишем запросы, выданные преподавателем на защите прошлой работы:

1. Вывести название клуба, его страну, тренера, средний показатель голов за игру, количество побед:

```
EXPLAIN ANALYSE SELECT cl.name as "ClubName", co.name as
"Country", tr.firstname as "Trainer", hw.wh + gw.wg as "wins", (hg.goc +
gg.goc)* 1.0 / (hg.gac + gg.gac) as "AvgGoalPerGame"
from clubs cl
JOIN cities ci on cl.cityId = ci.id
JOIN countries co ON ci.countryid = co.id
JOIN trainers tr on cl.id = tr.clubId
JOIN (SELECT homeClubId, COUNT(*) as "wh" FROM games WHERE
homeTeamScore > guestTeamScore group by homeClubId) as hw ON
hw.homeClubId = cl.id
JOIN (SELECT guestClubId, COUNT(*) as "wg" FROM games WHERE
homeTeamScore < guestTeamScore group by guestClubId) as gw ON
gw.guestClubId = cl.id
JOIN (SELECT homeClubId, SUM(homeTeamScore) as "goc", COUNT(*)
as "gac" FROM games GROUP BY homeClubId) as hg on hg.homeClubId
= cl.id
JOIN (SELECT guestClubId, SUM(guestTeamScore) as "goc", COUNT(*)
as "gac" FROM games GROUP BY guestClubId) as gg on gg.guestClubId =
cl.id;
```

2. Вывести топ игроков по голам и процент побед игрока:

```
EXPLAIN ANALYSE SELECT pc.firstname as "FirstName", pg.goals as
"Goals", (hw.wh + gw.wg) / (hg.games + gg.games) as "PercentOfWins"
FROM (SELECT scorerPassport, COUNT(*) as "goals" FROM goals group
by scorerPassport) as pg
JOIN (SELECT firstname, passport, clubId FROM Players) as pc ON
pg.scorerPassport = pc.passport
JOIN (SELECT homeClubId, COUNT(*) as "wh" FROM games WHERE
homeTeamScore > guestTeamScore group by homeClubId) as hw ON
hw.homeClubId = pc.clubId
JOIN (SELECT guestClubId, COUNT(*) as "wg" FROM games WHERE
homeTeamScore < guestTeamScore group by guestClubId) as gw ON
gw.guestClubId = pc.clubId
```

- ```

JOIN (SELECT homeClubId, COUNT(*) as "games" FROM games
GROUP BY homeClubId) as hg on hg.homeClubId = pc.clubId
JOIN (SELECT guestClubId, COUNT(*) as "games" FROM games
GROUP BY guestClubId) as gg on gg.guestClubId = pc.clubId
ORDER BY pg.goals;

```
3. Вывести средний показатель удалений игрока за игру:  

```

EXPLAIN ANALYSE SELECT pg.playerPassport, pc.firstname, pg.time /
(hg.games + gg.games) as "Time"
FROM (SELECT playerPassport, SUM(penaltyTime) as "time" FROM
penalties group by playerPassport) as pg
JOIN (SELECT firstname, passport, clubId FROM Players) as pc ON
pg.playerPassport = pc.passport
JOIN (SELECT homeClubId, COUNT(*) as "games" FROM games
GROUP BY homeClubId) as hg on hg.homeClubId = pc.clubId
JOIN (SELECT guestClubId, COUNT(*) as "games" FROM games
GROUP BY guestClubId) as gg on gg.guestClubId = pc.clubId;

```
  4. Вывести количество игроков в каждой стране:  

```

EXPLAIN ANALYSE SELECT countries.name, sm.pc as "Count"
FROM Countries as countries
JOIN (SELECT countryId, COUNT(passport) as "pc"
FROM Players
GROUP BY countryId) as sm ON sm.countryId = countries.id;

```
  5. Вывести для каждого тренера его команду:  

```

EXPLAIN ANALYSE SELECT tr.firstname, tr.lastname, cl.name
FROM Trainers as tr
JOIN Clubs as cl ON tr.clubId = cl.id

```

## Анализ

Для анализа запросов прибавим к ним вначале слова «EXPLAIN ANALYSE». Также напишем скрипт, который будет N раз запускать каждый запрос, и считать средний, минимальный и максимальный показатели времени планирования и исполнения запроса:

```

#!/bin/bash
declare -A arr
calls=$(cat app/resources/docker-compose.yml | grep -i "EXPLAIN_CALLS" | tr
"=" " " | awk '{print $NF}')
z=${calls//[\r']}
file=""

```

```

print() {
 # echo "$file"
 echo -e "\tplan_time\ttexec_time" >> "$file"
 min="min"
 max="max"
 avg="avg"
 min="$min""\t""$(echo "$2" | awk '{printf "%.2f", $1}')"""\t""$(echo "$5" | awk
'{printf "%.2f", $1}')"
 max="$max""\t""$(echo "$3" | awk '{printf "%.2f", $1}')"""\t""$(echo "$6" | awk
'{printf "%.2f", $1}')"
 avg="$avg""\t""$(echo "$1" | awk '{printf "%.2f", $1}')"""\t""$(echo "$4" | awk
'{printf "%.2f", $1}')"
 echo -e $min >> "$file"
 echo -e $max >> "$file"
 echo -e $avg >> "$file"
 echo "_____ " >> "$file"
}

```

```

while [[$i -le 5]]; do
 max_plan=-1
 min_plan=1000000000000000
 avg_plan=0
 max_exec=-1
 min_exec=1000000000000000
 avg_exec=0
 j=1
 while [[$j -le "$z"]]; do
 s=$(bash app/analyze/ex$i.sh)
 p=$(echo "$s" | grep -i "Planning")
 d=$(echo "$s" | grep -i "Execution")
 #newline=$(echo "$line" | tr "=" " ")
 read fw sw tw qw <<< $p
 avg_plan=$(awk -v cur="$tw" -v cmp="$avg_plan" 'BEGIN {print cmp +
cur; }')
 max_plan=$(awk -v cur="$tw" -v cmp="$max_plan" 'BEGIN {if (cur >=
cmp) print cur; else print cmp; }')
 min_plan=$(awk -v cur="$tw" -v cmp="$min_plan" 'BEGIN {if (cur <=
cmp) print cur; else print cmp; }')
 done
done

```

```

read fw sw tw qw <<< $d

avg_exec=$(awk -v cur="$tw" -v cmp="$avg_exec" 'BEGIN {print cmp +
cur; }')
max_exec=$(awk -v cur="$tw" -v cmp="$max_exec" 'BEGIN {if (cur >=
cmp) print cur; else print cmp; }')
min_exec=$(awk -v cur="$tw" -v cmp="$min_exec" 'BEGIN {if (cur <=
cmp) print cur; else print cmp; }')

((j++))
done
avg_exec=$(awk -v sum="$avg_exec" -v count="$z" 'BEGIN { print sum /
count; }')
avg_plan=$(awk -v sum="$avg_plan" -v count="$z" 'BEGIN { print sum /
count; }')
file="app/analyze/Result_""$i"
print "$avg_plan" "$min_plan" "$max_plan" "$avg_exec" "$min_exec"
"$max_exec"
echo "Query "$i"/5"
((i++))
done
echo Done!

```

## Индексы

Попробуем проиндексировать следующие поля:

```

#!/bin/bash
db="$1"
user="$2"
port="$3"
password="$4"
SQL="
CREATE INDEX IF NOT EXISTS idx_homeclubid ON Games USING btree
(homeClubId);
"
PGPASSWORD="$password" psql -U "$user" -d "$db" -p "$port" -v
ON_ERROR_STOP=1 -c "$SQL" --quiet
SQL="

```

```
CREATE INDEX IF NOT EXISTS idx_guestclubid ON Games USING btree
(guestClubId);
```

"

```
PGPASSWORD="$password" psql -U "$user" -d "$db" -p "$port" -v
ON_ERROR_STOP=1 -c "$SQL" --quiet
SQL="
```

```
CREATE INDEX IF NOT EXISTS idx_trainerclubid ON trainers USING btree
(clubId);
```

"

```
PGPASSWORD="$password" psql -U "$user" -d "$db" -p "$port" -v
ON_ERROR_STOP=1 -c "$SQL" --quiet
SQL="
```

```
CREATE INDEX IF NOT EXISTS idx_cityid ON Clubs USING btree (cityId);
```

"

```
PGPASSWORD="$password" psql -U "$user" -d "$db" -p "$port" -v
ON_ERROR_STOP=1 -c "$SQL" --quiet
SQL="
```

```
CREATE INDEX IF NOT EXISTS idx_scorerpassport ON goals USING btree
(scorerPassport);
```

"

```
PGPASSWORD="$password" psql -U "$user" -d "$db" -p "$port" -v
ON_ERROR_STOP=1 -c "$SQL" --quiet
SQL="
```

```
CREATE INDEX IF NOT EXISTS idx_playercountry ON players USING btree
(countryId);
```

"

```
PGPASSWORD="$password" psql -U "$user" -d "$db" -p "$port" -v
ON_ERROR_STOP=1 -c "$SQL" --quiet
```

В результате большинство запросов стало работать хуже, чем первоначально. Это может быть связано с тем, что в запросах мы проходим все поля таблицы, а не ищем какие-то уникальные значения.

| Type | Start  | Indexes | penaltyIndex | scorerIndex | countyIndex |
|------|--------|---------|--------------|-------------|-------------|
| min  | 7,5    | 6,74    | 7,64         | 6,86        | 7,88        |
| max  | 12,77  | 10      | 9,29         | 8,81        | 11,48       |
| avg  | 9,53   | 7,81    | 8,48         | 8,01        | 9,64        |
| min  | 600,64 | 412,62  | 423,25       | 447,26      | 454,9       |
| max  | 641,17 | 434,26  | 512,93       | 465,09      | 561,15      |
| avg  | 613,08 | 426,28  | 454,93       | 455,24      | 502,09      |
| min  | 2,65   | 2,08    | 2,21         | 2,29        | 2,22        |
| max  | 7,62   | 3,71    | 4,72         | 4,38        | 4,16        |

|     |          |          |          |          |          |
|-----|----------|----------|----------|----------|----------|
| avg | 4,79     | 3,13     | 3,33     | 3,33     | 3,26     |
| min | 56207,61 | 51313,93 | 52223,64 | 50017,24 | 50393,94 |
| max | 76963,83 | 52407,58 | 52034,2  | 52034,2  | 50880,83 |
| avg | 69011,8  | 52032,2  | 54204,2  | 50786,6  | 50678    |
| min | 1,05     | 1,09     | 1,35     | 0,93     | 1,05     |
| max | 1,96     | 2,21     | 3,11     | 1,21     | 1,76     |
| avg | 1,38     | 1,55     | 2,01     | 1,03     | 1,5      |
| min | 10051,09 | 9969,34  | 41867,01 | 10247,55 | 9549,91  |
| max | 12352,7  | 12945,59 | 48657,7  | 1179,02  | 10748,6  |
| avg | 10750,7  | 11127,5  | 44593,2  | 10516,4  | 10185,4  |
| min | 0,62     | 0,88     | 0,73     | 0,58     | 0,78     |
| max | 1,57     | 1,76     | 1,5      | 1,9      | 2,21     |
| avg | 1,03     | 1,4      | 1,13     | 1,01     | 1,13     |
| min | 176,27   | 160,29   | 167,38   | 160,17   | 166,44   |
| max | 236,27   | 218,5    | 309,28   | 219,11   | 223,89   |
| avg | 195,5    | 199,65   | 237,24   | 186,16   | 185,32   |
| min | 0,86     | 1,12     | 0,84     | 0,88     | 1,02     |
| max | 1,96     | 2,69     | 2,06     | 1,25     | 2,94     |
| avg | 1,18     | 1,47     | 1,43     | 1,1      | 1,51     |
| min | 0,81     | 0,92     | 0,72     | 0,7      | 0,83     |
| max | 2,04     | 1,77     | 2,41     | 0,96     | 1,86     |
| avg | 1,22     | 1,37     | 1,37     | 0,88     | 1,25     |

## Партиционирование

Разделим таблицу penalties на партии. Принципом деления будет служить номер паспорта игрока. Всего паспорт бывает в диапазоне от 1 до 1000000, потому поделим таблицу на 10 партий (playerPassport в отрезках 1..100000, 100001..200000 и так далее):

```
#!/bin/bash
```

```
SQL="
```

```
CREATE TABLE IF NOT EXISTS penalties_new (
 gameTime time,
 gameId int,
 playerPassport bigint,
 penaltyTime time
) PARTITION BY RANGE (playerPassport);
"
```

```
psql -h localhost -U postgres -d dbhw2 -p 5432 -v ON_ERROR_STOP=1 -c
"$SQL" --quiet
```

```
SQL="
```

```
CREATE TABLE IF NOT EXISTS playerPassport_1 PARTITION OF
penalties_new FOR VALUES FROM (1) TO (100001);
```

```
CREATE TABLE IF NOT EXISTS playerPassport_2 PARTITION OF
penalties_new FOR VALUES FROM (100001) TO (200001);
```

```
CREATE TABLE IF NOT EXISTS playerPassport_3 PARTITION OF
penalties_new FOR VALUES FROM (200001) TO (300001);
```

```
CREATE TABLE IF NOT EXISTS playerPassport_4 PARTITION OF
penalties_new FOR VALUES FROM (300001) TO (400001);
```

```
CREATE TABLE IF NOT EXISTS playerPassport_5 PARTITION OF
penalties_new FOR VALUES FROM (400001) TO (500001);
```

```
CREATE TABLE IF NOT EXISTS playerPassport_6 PARTITION OF
penalties_new FOR VALUES FROM (500001) TO (600001);
```

```
CREATE TABLE IF NOT EXISTS playerPassport_7 PARTITION OF
penalties_new FOR VALUES FROM (600001) TO (700001);
```

```
CREATE TABLE IF NOT EXISTS playerPassport_8 PARTITION OF
penalties_new FOR VALUES FROM (700001) TO (800001);
```

```
CREATE TABLE IF NOT EXISTS playerPassport_9 PARTITION OF
penalties_new FOR VALUES FROM (800001) TO (900001);
```

```
CREATE TABLE IF NOT EXISTS playerPassport_10 PARTITION OF
penalties_new FOR VALUES FROM (900001) TO (1000001);
```

"

```
psql -h localhost -U postgres -d dbhw2 -p 5432 -v ON_ERROR_STOP=1 -c
"$SQL" --quiet
```

Мы создали новую таблицу, которую разделили на 10 партиций. Далее перенесем данные из старой таблицы в новую запросом:

```
INSERT INTO penalties_new SELECT * FROM penalties;
```

Далее переименуем таблицы и подвяджем внешние ключи:

```
ALTER TABLE penalties RENAME TO old_penalties;
```

```
ALTER TABLE penalties_new RENAME TO penalties;
```

```
ALTER TABLE playerpassport_1 ADD CONSTRAINT fk_playerPassport1
FOREIGN KEY (playerpassport) REFERENCES Players (passport);
```

```
ALTER TABLE playerpassport_2 ADD CONSTRAINT fk_playerPassport2
FOREIGN KEY (playerpassport) REFERENCES Players (passport);
```

```
ALTER TABLE playerpassport_3 ADD CONSTRAINT fk_playerPassport3
FOREIGN KEY (playerpassport) REFERENCES Players (passport);
```

```
ALTER TABLE playerpassport_4 ADD CONSTRAINT fk_playerPassport4
FOREIGN KEY (playerpassport) REFERENCES Players (passport);
```

```
ALTER TABLE playerpassport_5 ADD CONSTRAINT fk_playerPassport5
FOREIGN KEY (playerpassport) REFERENCES Players (passport);
```



```
ALTER TABLE playerpassport_6 ADD CONSTRAINT fk_playerPassport6
FOREIGN KEY (playerpassport) REFERENCES Players (passport);
ALTER TABLE playerpassport_7 ADD CONSTRAINT fk_playerPassport7
FOREIGN KEY (playerpassport) REFERENCES Players (passport);
ALTER TABLE playerpassport_8 ADD CONSTRAINT fk_playerPassport8
FOREIGN KEY (playerpassport) REFERENCES Players (passport);
ALTER TABLE playerpassport_9 ADD CONSTRAINT fk_playerPassport9
FOREIGN KEY (playerpassport) REFERENCES Players (passport);
ALTER TABLE playerpassport_10 ADD CONSTRAINT fk_playerPassport10
FOREIGN KEY (playerpassport) REFERENCES Players (passport);
```

```
ALTER TABLE playerpassport_1 ADD CONSTRAINT fk_gameId1
FOREIGN KEY (gameId) REFERENCES Games (id);
ALTER TABLE playerpassport_2 ADD CONSTRAINT fk_gameId2
FOREIGN KEY (gameId) REFERENCES Games (id);
ALTER TABLE playerpassport_3 ADD CONSTRAINT fk_gameId3
FOREIGN KEY (gameId) REFERENCES Games (id);
ALTER TABLE playerpassport_4 ADD CONSTRAINT fk_gameId4
FOREIGN KEY (gameId) REFERENCES Games (id);
ALTER TABLE playerpassport_5 ADD CONSTRAINT fk_gameId5
FOREIGN KEY (gameId) REFERENCES Games (id);
ALTER TABLE playerpassport_6 ADD CONSTRAINT fk_gameId6
FOREIGN KEY (gameId) REFERENCES Games (id);
ALTER TABLE playerpassport_7 ADD CONSTRAINT fk_gameId7
FOREIGN KEY (gameId) REFERENCES Games (id);
ALTER TABLE playerpassport_8 ADD CONSTRAINT fk_gameId8
FOREIGN KEY (gameId) REFERENCES Games (id);
ALTER TABLE playerpassport_9 ADD CONSTRAINT fk_gameId9
FOREIGN KEY (gameId) REFERENCES Games (id);
ALTER TABLE playerpassport_10 ADD CONSTRAINT fk_gameId10
FOREIGN KEY (gameId) REFERENCES Games (id);
```

И добавим первичные ключи

SQL="

```
ALTER TABLE playerpassport_1 ADD PRIMARY KEY (gameTime, gameId,
playerPassport);
ALTER TABLE playerpassport_2 ADD PRIMARY KEY (gameTime, gameId,
playerPassport);
ALTER TABLE playerpassport_3 ADD PRIMARY KEY (gameTime, gameId,
playerPassport);
```

ALTER TABLE playerpassport\_4 ADD PRIMARY KEY (gameTime, gameId, playerPassport);

ALTER TABLE playerpassport\_5 ADD PRIMARY KEY (gameTime, gameId, playerPassport);

ALTER TABLE playerpassport\_6 ADD PRIMARY KEY (gameTime, gameId, playerPassport);

ALTER TABLE playerpassport\_7 ADD PRIMARY KEY (gameTime, gameId, playerPassport);

ALTER TABLE playerpassport\_8 ADD PRIMARY KEY (gameTime, gameId, playerPassport);

ALTER TABLE playerpassport\_9 ADD PRIMARY KEY (gameTime, gameId, playerPassport);

ALTER TABLE playerpassport\_10 ADD PRIMARY KEY (gameTime, gameId, playerPassport);

"

PGPASSWORD="\$password" psql -U "\$user" -d "\$db" -p "\$port" -v  
ON\_ERROR\_STOP=1 -c "\$SQL" -quiet

Также была попытка добавить отдельные индексы к некоторым столбцам, но при этом мы проиграли в производительности. Итоговые замеры:

| Запрос | Time | Type | Partitions | Primary_key | Indexes  |
|--------|------|------|------------|-------------|----------|
| 1      | plan | min  | 6,91       | 8,43        | 8,8      |
|        |      | max  | 10,33      | 11,73       | 14,53    |
|        |      | avg  | 9,05       | 9,56        | 10,81    |
|        | exec | min  | 456,41     | 463,15      | 494,7    |
|        |      | max  | 498,98     | 496,24      | 555,06   |
|        |      | avg  | 476,67     | 482,62      | 512,57   |
| 2      | plan | min  | 2,11       | 2,07        | 2,23     |
|        |      | max  | 4,87       | 4,28        | 3,33     |
|        |      | avg  | 3,3        | 2,86        | 2,63     |
|        | exec | min  | 51800,86   | 53731,3     | 54129,99 |
|        |      | max  | 53892,84   | 55083,534   | 70184,96 |
|        |      | avg  | 52594,2    | 54522,8     | 58387,4  |
| 3      | plan | min  | 1,61       | 2,24        | 5,23     |
|        |      | max  | 5,54       | 4,5         | 18,2     |
|        |      | avg  | 3,04       | 3,36        | 11,2     |
|        | exec | min  | 11501,68   | 10767,97    | 13101,33 |
|        |      | max  | 13813,47   | 11939,08    | 17889,93 |
|        |      | avg  | 12518,4    | 11280       | 14666,5  |
| 4      | plan | min  | 0,75       | 0,64        | 0,69     |
|        |      | max  | 1          | 0,97        | 2,69     |
|        |      | avg  | 0,89       | 0,81        | 1,63     |
|        | exec | min  | 169,08     | 177,53      | 173,06   |
|        |      | max  | 244,24     | 298,76      | 225,5    |

|   |      |     |        |        |        |
|---|------|-----|--------|--------|--------|
|   |      | avg | 204,69 | 216,97 | 200,91 |
| 5 | plan | min | 0,96   | 1,04   | 0,87   |
|   |      | max | 2,52   | 1,24   | 1,39   |
|   |      | avg | 1,58   | 1,13   | 1,11   |
|   | exec | min | 0,76   | 0,75   | 0,65   |
|   |      | max | 2,34   | 2,71   | 2,6    |
|   |      | avg | 1,24   | 1,26   | 1,23   |

В большинстве запросов время выполнения выросло. В 3 запросе, который работает с партиционированной таблицей, скачка нет, однако запрос все-таки стал работать дольше, хотя и использует созданные партии (группировка в них быстрее).

## Бэкапы

Для бэкапов напишем такой скрипт:

```
mxcount=$(cat old-docker-compose.yml | grep BACKUPS_COPIES | tr "=" " " |
awk '{print $NF}')
```

```
freq=$(cat old-docker-compose.yml | grep BACKUPS_FREQUENCY | tr "=" " " |
awk '{print $NF}')
```

```
f=$(date +%H)
```

```
curr=$(cat backup/curr.txt)
```

```
count=$current_file
```

```
deletenum=0
```

```
#PGPASSWORD="1234" pg_dump -U kosandron -d postgres -p 5000 -f
backup/backup"$curr".sql
```

```
PGPASSWORD="1234" pg_dump -U kosandron -d postgres -p 5000 -f
backup/backup"$curr".sql
```

```
deletenum=$(awk -v now="$curr" -v mx="$mxcount" 'BEGIN {print now - mx }')
```

```
if [-f backup$deletenum.sql]; then
```

```
 rm backup$deletenum.sql
```

```
fi
```

```
((curr++))
```

```
echo "$curr" > backup/curr.txt
```

```
while true; do
```

```
 s=$(date +%H)
```

```
 dif=$(awk -v last="$f" -v now="$s" 'BEGIN {print now - last}')
```

```
 if [["$dif" -ge "0"]]; then
```

```

if [["$dif" -ge "$freq"]]; then

 PGPASSWORD="1234" pg_dump -U kosandron -d postgres -p 5000 -f
 backup/backup"$curr".sql

 deletenum=$(awk -v now="$curr" -v mx="$mxcount" 'BEGIN {print now
- mx }')

 if [-f backup$deletenum.sql]; then

 rm backup$deletenum.sql

 fi

 ((curr++))

 echo "$curr" > backup/curr.txt

 f=$s

fi

else

 let temp=$s+24

 dif=$(awk -v last="$f" -v now="$temp" 'BEGIN {print now - last}')

 if [["$dif" -ge "$freq"]]; then

 PGPASSWORD="1234" pg_dump -U kosandron -d postgres -p 5000 -f
 backup/backup"$curr".sql

 deletenum=$(awk -v now="$curr" -v mx="$mxcount" 'BEGIN {print now
- mx }')

 if [-f backup$deletenum.sql]; then

 rm backup$deletenum.sql

 fi

 ((curr++))

 echo "$curr" > backup/curr.txt

```

f=\$s

fi

fi





sleep 1

done

## Отказоустойчивость

Развернем patroni в докер-контейнере:

По итогу видим, что одна партиция является лидером, а другая ее фолловером:

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |         |             |      |       |       |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|-------------|------|-------|-------|
| <div><div><div>&lt;</div><div></div><div>patroni1</div></div><div><a href="#">patroni</a></div><div>b04d3caa12a7 </div></div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |         |             |      |       |       |
| Logs                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Inspect | Bind mounts | Exec | Files | Stats |
| <div>2024-05-20 13:20:31 2024-05-20 10:20:31,178 INFO: no action. I am (patroni1), the leader with the lock</div> <div>2024-05-20 13:20:41 2024-05-20 10:20:41,138 INFO: no action. I am (patroni1), the leader with the lock</div> <div>2024-05-20 13:20:50 2024-05-20 10:20:50,994 INFO: no action. I am (patroni1), the leader with the lock</div> <div>2024-05-20 13:21:01 2024-05-20 10:21:01,196 INFO: no action. I am (patroni1), the leader with the lock</div> <div>2024-05-20 13:21:11 2024-05-20 10:21:11,144 INFO: no action. I am (patroni1), the leader with the lock</div> <div>2024-05-20 13:21:21 2024-05-20 10:21:21,035 INFO: no action. I am (patroni1), the leader with the lock</div> <div>2024-05-20 13:21:31 2024-05-20 10:21:31,156 INFO: no action. I am (patroni1), the leader with the lock</div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |         |             |      |       |       |
| <div><div><div>&lt;</div><div></div><div>patroni2</div></div><div><a href="#">patroni</a></div><div>105998d7247d </div></div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |         |             |      |       |       |
| Logs                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Inspect | Bind mounts | Exec | Files | Stats |
| <div>2024-05-20 13:20:51 2024-05-20 10:20:51,541 INFO: no action. I am (patroni2), a secondary, and following a leader (patroni1)</div> <div>2024-05-20 13:21:01 2024-05-20 10:21:01,786 INFO: no action. I am (patroni2), a secondary, and following a leader (patroni1)</div> <div>2024-05-20 13:21:11 2024-05-20 10:21:11,680 INFO: no action. I am (patroni2), a secondary, and following a leader (patroni1)</div> <div>2024-05-20 13:21:21 2024-05-20 10:21:21,583 INFO: no action. I am (patroni2), a secondary, and following a leader (patroni1)</div> <div>2024-05-20 13:21:31 2024-05-20 10:21:31,757 INFO: no action. I am (patroni2), a secondary, and following a leader (patroni1)</div> <div>2024-05-20 13:21:41 2024-05-20 10:21:41,657 INFO: no action. I am (patroni2), a secondary, and following a leader (patroni1)</div> <div>2024-05-20 13:21:51 2024-05-20 10:21:51,701 INFO: no action. I am (patroni2), a secondary, and following a leader (patroni1)</div> <div>2024-05-20 13:22:01 2024-05-20 10:22:01,695 INFO: no action. I am (patroni2), a secondary, and following a leader (patroni1)</div> <div>2024-05-20 13:22:11 2024-05-20 10:22:11,791 INFO: no action. I am (patroni2), a secondary, and following a leader (patroni1)</div> <div>2024-05-20 13:22:21 2024-05-20 10:22:21,585 INFO: no action. I am (patroni2), a secondary, and following a leader (patroni1)</div> |         |             |      |       |       |

А при отключении одного из серверов лидером станет другой.