

PG BootCamp Russia 2024 Kazan



PGBootCamp.ru

Лайфхаки от DBA

Александр Никитин, ведущий администратор БД PGMechanix



Наша команда PostgreSQL DBA

Работаем с Big Data

Поддерживаем **тысячи БД** объемом
свыше **30 ТБ** на хост.

Поддерживаем mission-critical БД

Обеспечиваем доступность команды
24/7/365.

SLA на все обращения

Гарантированные сроки реакции на
обращения. SLA на аварии **от 30 мин.**

Делимся знаниями

За плечами у каждого
не менее **10 лет** опыта
работы с PostgreSQL.

Занимаемся разработкой

Патчи наших DBA внесены в
mainstream код PostgreSQL.

Проводим диагностику

Нашими DBA подано около
300 bug reports.



Наши клиенты

платежные системы



разработка

банки



ритейл

телеком



геймдев

логистика



образование







Чем ты занимаешься после работы?



Я PostgreSQL DBA...



⚙ Инструменты

⚙ Примеры использования



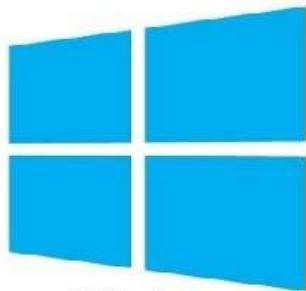
- git

- git
- `git clone https://github.com/dataegret/pg-utils ~/stuff`



- <https://www.youtube.com/watch?v=qJf3UmD4HFk>
- Инструменты от Data Egret для облегчения администрирования БД.
PGDay 2021





Windows



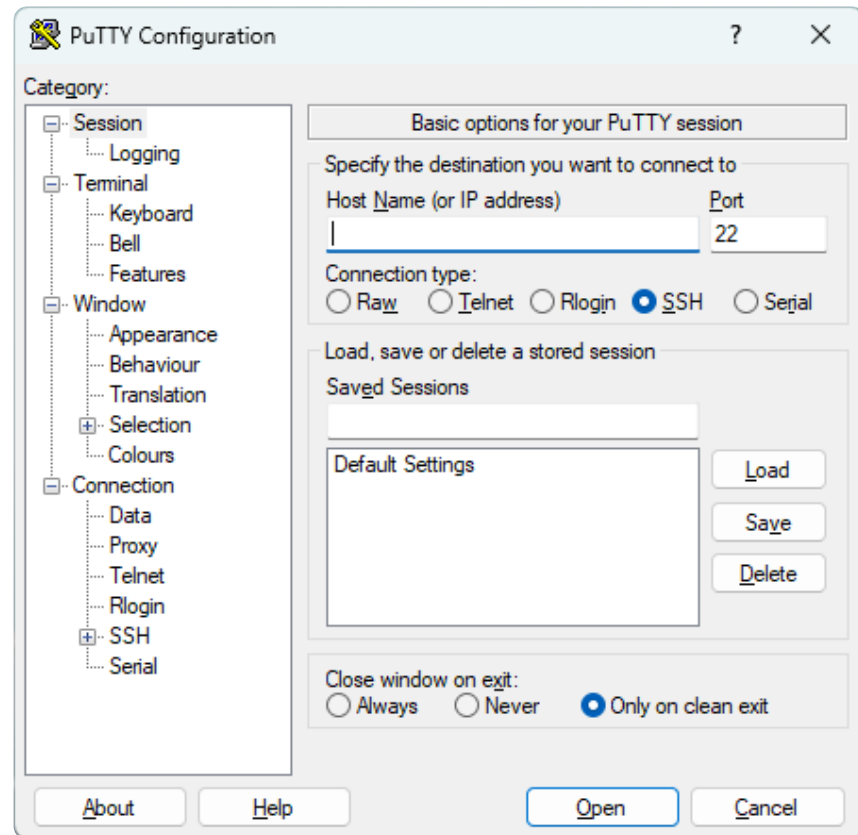
Mac



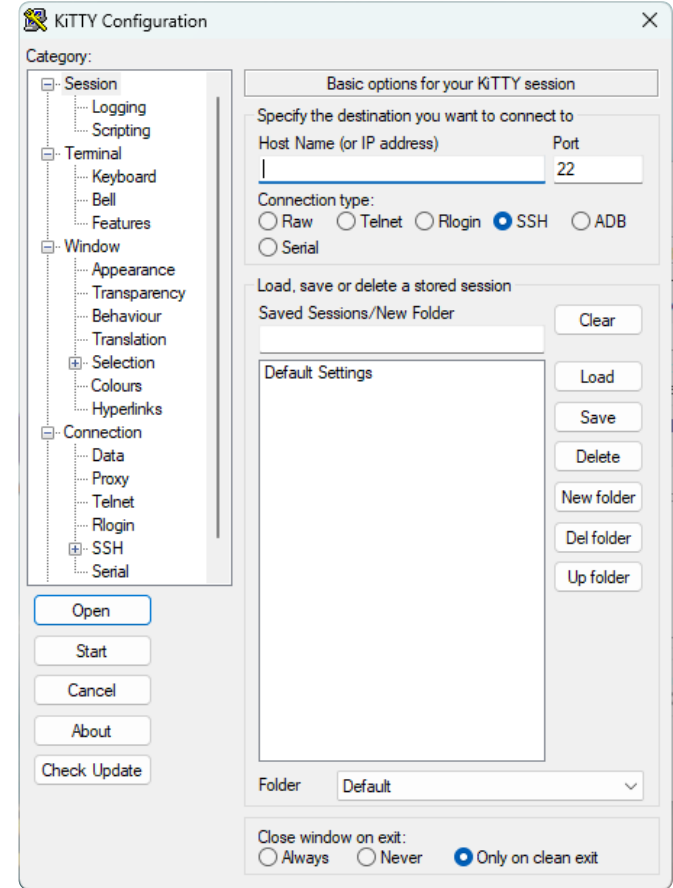
Linux



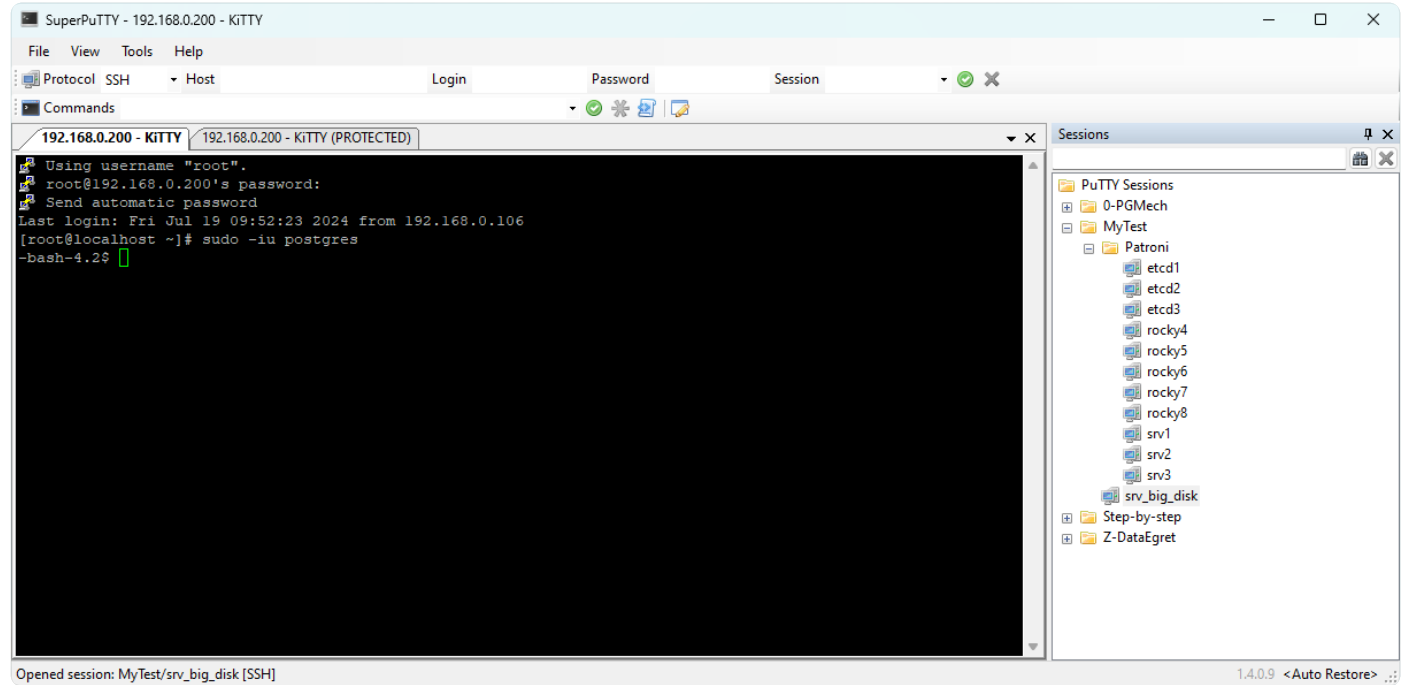
- PuTTY (<https://www.putty.org/>)



- PuTTY (<https://www.putty.org/>)
- Kitty (<http://www.9bis.net/kitty/#!index.md>)



- SuperPutty (<https://superputty.org/>)



- Tools — Options
 - Putty.exe location = kitty.exe

- Tools — Options
 - Putty.exe location = kitty.exe
- Ctrl+F9

- Tools — Options
 - Putty.exe location = kitty.exe
- Ctrl+F9
- Отправка одной команды в несколько терминалов

- AutoHotKey (<https://www.autohotkey.com/>)



- AutoHotKey (<https://www.autohotkey.com/>)
- Игры :)



- Alt+4 (Вход под пользователем OS postgres)

Alt & 4::

```
temp := clipboard  
clipboard := "sudo -iu postgres"  
Send, +{Ins}{Enter}  
clipboard := temp  
return
```

- Alt+1 (Что происходит в PostgreSQL)

Alt & 1::

```
temp := clipboard  
clipboard := "\i ~/stuff/sql/db_activity9.6.sql"  
Send, +{Ins}{Enter}  
clipboard := temp  
return
```

- Alt+1 (Что происходит в PostgreSQL)

```
postgres=# \x
Expanded display is on.
postgres=# \i ~/stuff/sql/db_activity9.6.sql
-[ RECORD 1 ]-----
ts_age      | 00:01:21.217617
state       | active
query_age   | 00:00:35.822871
change_age  | 00:00:35.822867
datname     | pricing
pid         | 22911
username    | hive
waiting     | f
client_addr |
client_port | -1
query       | SELECT "black_price","red_price","sku","country","start_date","end_date","source","modified_by","created_at","actions_info" FROM (+
          | SELECT *
          | FROM prices
          | WHERE created_at >= '2022-07-12 00:00:00.0'
          | AND created_at < '2022-07-19 00:00:00.0'
          | ) jdbc_query WHERE "created_at" >= '2022-07-17 06:00:00'
-[ RECORD 2 ]-----
```

- Alt+2 (Находится ли кластер в режиме восстановления)

Alt & 2::

```
temp := clipboard  
clipboard := "select pg_is_in_recovery();"   
Send, +{Ins}{Enter}  
clipboard := temp  
return
```

- Ctrl+2 (Состояние реплик)

^2::

temp := clipboard

clipboard := "SELECT client_addr AS client... FROM pg_stat_replication;"

Send, +{Ins}{Enter}

clipboard := temp

return

- Ctrl+2 (Состояние реплик)

```
-[ RECORD 1 ]+-----  
client      | 192.168.0.164  
user        | replica  
name        | rocky4  
state       | streaming  
mode        | async  
backend_xmin |  
pending_mb  | 0.0  
write_mb    | 0.0  
flush_mb    | 0.0  
replay_mb   | 0.0  
total_mb    | 0.0  
replay_lag  |
```

pending - уже сгенерирован, на мастере, но не отправлен (скорее всего проблема на мастере - он по каким-то причинам не может отправить данные на реплику, перегружены диски или не работает сеть)

write - позиция журнала транзакций, отправленного на реплику, то есть оно не доехало до реплики (только сеть)

flush - позиция журнала транзакций, записанного на мастер (проблема на реплике)

replay - сброшенный на диск (проблема на реплике)

total_lag - воспроизведённые данные

- Alt+3 (Дерево блокировок)

Alt & 3::

```
temp := clipboard  
clipboard := "\i ~/stuff/sql/locktree.sql"  
Send, +{Ins}{Enter}  
clipboard := temp  
return
```

- Alt+3 (Дерево блокировок)

```
postgres=# \i ~/stuff/sql/locktree.sql
 ts_age | change_age | datname | username | client_addr | pid | state | lvl | blocked | query
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
 00:00:57 | 00:00:40 | postgres | postgres |              | 3244 | idletx | 0 | 1 | update big set content = 'aaa' where id1 = 1;
 00:00:30 | 00:00:08 | postgres | postgres |              | 3301 | active | 1 | 0 | . update big set content = 'bbb' where id1 = 1;
(2 rows)
```

- ctrl+5 (explain (analyze, buffers))

^5::

temp := clipboard

clipboard := "explain (analyze, buffers)"

Send, +{Ins}{Space}

clipboard := temp

return

- Alt+5 (Безопасный explain (analyze, buffers))

Alt & 5::

```
temp := clipboard  
clipboard := "begin; set  idle_in_transaction_session_timeout='10s'; explain (analyze, buffers)  
rollback;"  
Send, +{Ins}{Left 11}  
clipboard := temp  
return
```

- Alt+6 (Статистика по полям)

Alt & 6::

temp := clipboard

clipboard := "select * from pg_stats where tablename = " and attname = " \gx"

Send, +{Ins}{Left 22}

clipboard := temp

return

- Alt+6 (Статистика по полям)

```
schemaname      | public
tablename       | big
attname         | id2
inherited       | f
null_frac       | 0
avg_width       | 4
n_distinct      | -1
most_common_vals |
most_common_freqs |
histogram_bounds | {2,19832,40505,59742,79854,98601,117258,13670
440518,461484,481244,502469,520393,539744,560192,577263,595658,616273,
1553,940041,959406,978784,1000048,1020037,1041110,1060766,1080335,1100
1360044,1379473,1400716,1423898,1443554,1464461,1485663,1505118,152586
80187,1799180,1816594,1836964,1856820,1878233,1898170,1916917,1938703,
correlation     | -0.31773466
most_common_elems |
most_common_elem_freqs |
elem_count_histogram |
```

- Alt+7 (Переход к логам)

Alt & 7::

```
temp := clipboard  
clipboard := "cd /var/log/postgresql"  
Send, +{Ins}{Enter}  
clipboard := "ls"  
Send, +{Ins}{Enter}  
clipboard := temp  
return
```

- Alt+8 (Список сервисов|grep postgres)

Alt & 8::

temp := clipboard

clipboard := "systemctl list-units -t service --all|grep postgres"

Send, +{Ins}{Enter}

clipboard := temp

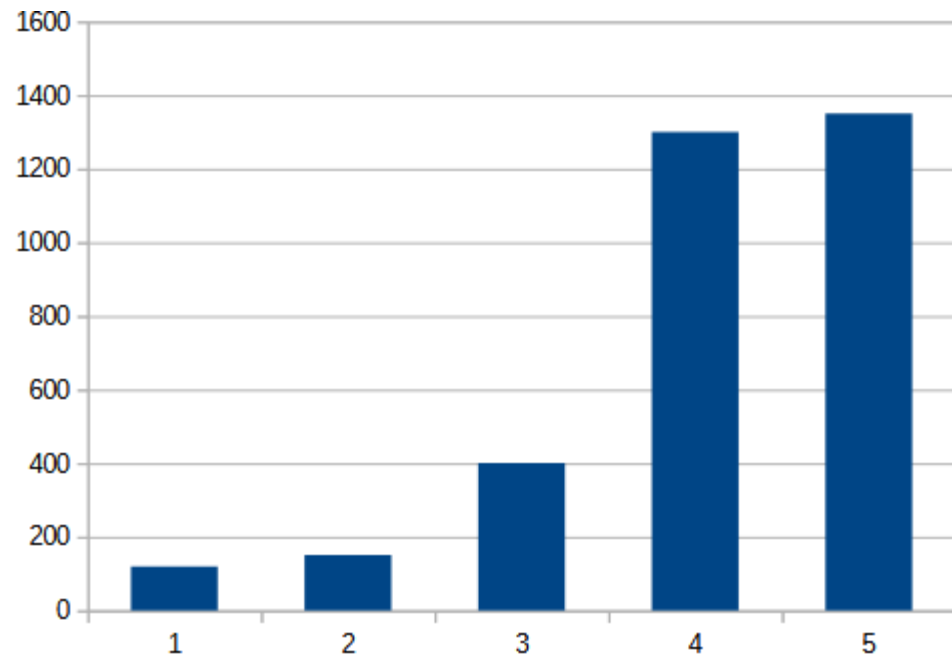
return

- <https://github.com/Nikitin-Alexandr/utils/blob/main/autoHotKey.ahk>



- Ну и не будем уходить далеко, от sed, vim, awk и прочих.





#Задаём имя файла

```
export fname=postgresql-2024-05-30.log
```

#Вытаскиваем часы, которые встречаются в этом файле в строку

```
h_str=""
```

#Если нет пропусков по часам, (к примеру с 13 до 14 не было записей), то строку с часами генерируем так:

```
for ((i=0; i<=23; i++ )) do
```

```
  h_str="${h_str} `printf "%0*d\n " 2 $i;`"
```

```
done
```

#Если в файле могут быть пропуски по времени, то так:

```
#h_str=`echo $(grep -E "^${fname:11:10}" $fname|cut -c 12-13|uniq) |tr '\n' ' '`
```

```
#Создаём массив, для вытащенных часов
arr=()
#Заполняем его
for i in $h_str; do arr+=($i) ; done
#Формируем команды
for ((i=0; i<${#arr[@]}-1; i++ )); do
    echo "echo -n '${arr[i]}: ' && sed -n '/^${fname:11:10} ${arr[i]}/,/^${fname:11:10} ${arr[i+1]}/p'
    $fname|wc -c"
done
echo "echo -n '${arr[i]}: ' && sed -n '/^${fname:11:10} ${arr[i]}/,\\$p' $fname|wc -c"
```

```
echo -n '00: ' && sed -n '/^2024-05-30 00/,/^2024-05-30 01/p' postgresql-2024-05-30.log|wc -c  
echo -n '01: ' && sed -n '/^2024-05-30 01/,/^2024-05-30 02/p' postgresql-2024-05-30.log|wc -c  
...  
echo -n '23: ' && sed -n '/^2024-05-30 23/,$p' postgresql-2024-05-30.log|wc -c
```

```
echo -n '00: ' && sed -n '/^2024-05-30 00/,/^2024-05-30 01/p' postgresql-2024-05-30.log|wc -c
echo -n '01: ' && sed -n '/^2024-05-30 01/,/^2024-05-30 02/p' postgresql-2024-05-30.log|wc -c
...
echo -n '23: ' && sed -n '/^2024-05-30 23/, $p' postgresql-2024-05-30.log|wc -c
```

Список команд:

export	cut
for	uniq
printf	tr
echo	sed
grep	wc

Пример вывода:

```
00: 40064561
01: 39643062
02: 44500692
03: 46424970
04: 56917609
05: 71193766
06: 98490940
07: 123821564
08: 145105797
09: 155717857
10: 159177827
11: 156302446
12: 154792428
13: 148677628
14: 140746099
15: 122423820
```

- 2024-09-06 00:00:00.686 MSK 105770 user@db_name from 10.13.19.81 [vxid:53/174763983 txid:0]
[SELECT] LOG: duration: 100.494 ms execute S_17: select totetransp0_.id as id1_1_, totetransp0_.created as created2_1_, totetransp0_.modified as modified3_1_, totetransp0_.version as version4_1_, totetransp0_.active_date as active_d5_1_, totetransp0_.complete_status as complete6_1_, totetransp0_.current_location_id as current15_1_, totetransp0_.destination_location_id as destina16_1_, totetransp0_.error_reason as error_re7_1_, totetransp0_.finished_date as finished8_1_, totetransp0_.forcedfinish as forcedfi9_1_, totetransp0_.number as number10_1_, totetransp0_.original_destination_location_id as origina17_1_, totetransp0_.source_location_id as source_18_1_, totetransp0_.status as status11_1_, totetransp0_.tote_number as tote_nu12_1_, totetransp0_.transport_batch_number as transpo13_1_, totetransp0_.type as type14_1_ from tote_transport_order totetransp0_ where totetransp0_.tote_number=\$1 and (totetransp0_.status in (\$2 , \$3 , \$4 , \$5 , \$6)) order by totetransp0_.modified desc limit \$7
- 2024-09-06 00:00:00.686 MSK 105770 user@db_name from 10.13.19.81 [vxid:53/174763983 txid:0]
[SELECT] DETAIL: parameters: \$1 = 'T00000005212', \$2 = 'CREATED', \$3 = 'ACTIVE', \$4 = 'FINISHED', \$5 = 'CANCELLED', \$6 = 'ERROR', \$7 = '1'

- 2024-09-06 00:00:00.686 MSK 105770 user@db_name from 10.13.19.81 [vxid:53/174763983 txid:0]
[SELECT] LOG: duration: 100.494 ms execute S_17: select totetransp0_.id as id1_1_, totetransp0_.created as created2_1_, totetransp0_.modified as modified3_1_, totetransp0_.version as version4_1_, totetransp0_.active_date as active_d5_1_, totetransp0_.complete_status as complete6_1_, totetransp0_.current_location_id as current15_1_, totetransp0_.destination_location_id as destina16_1_, totetransp0_.error_reason as error_re7_1_, totetransp0_.finished_date as finished8_1_, totetransp0_.forcedfinish as forcedfi9_1_, totetransp0_.number as number10_1_, totetransp0_.original_destination_location_id as origina17_1_, totetransp0_.source_location_id as source_18_1_, totetransp0_.status as status11_1_, totetransp0_.tote_number as tote_nu12_1_, totetransp0_.transport_batch_number as transpo13_1_, totetransp0_.type as type14_1_ from tote_transport_order totetransp0_ where totetransp0_.tote_number=\$1 and (totetransp0_.status in (\$2 , \$3 , \$4 , \$5 , \$6)) order by totetransp0_.modified desc limit \$7
- 2024-09-06 00:00:00.686 MSK 105770 user@db_name from 10.13.19.81 [vxid:53/174763983 txid:0]
[SELECT] DETAIL: parameters: \$1 = 'T00000005212', \$2 = 'CREATED', \$3 = 'ACTIVE', \$4 = 'FINISHED', \$5 = 'CANCELLED', \$6 = 'ERROR', \$7 = '1'

- https://github.com/Nikitin-Alexandr/Utils/blob/main/replace_parameters



```
CREATE OR REPLACE FUNCTION replace_parameters (query text, parameters text)
    RETURNS text
    AS $body$
DECLARE
    rec record;
    SQL text;
BEGIN
    SQL := '';
    FOR rec IN (
        SELECT
            array_to_string(REGEXP_MATCHES(parameters, '(?<=\$(\d+))(?=\=\ ', 'g'), ';') AS arg_no,
            array_to_string(REGEXP_MATCHES(parameters, '(?<=\= )(.*?)(?=(,\\ \$\d+|$\ ))', 'g'), ';') AS arg_val)
        LOOP
            SQL := regexp_replace(query, '\$ || rec.arg_no || '(?!\\d)', rec.arg_val, 'g');
            query := sql;
        END LOOP;
    RETURN sql;
END;
$body$
LANGUAGE 'plpgsql';
```

```
/*Example: select replace_parameters($$query$$,$$parameters$$);*/
```

Нужно найти все номера параметров:

```
array_to_string(REGEXP_MATCHES(parameters, '(?<=\$)(\d+)(?=\=\\ )', 'g'), ';') AS arg_no
```

lookahead - опережающая проверка - $X(?=Y)$ - найти X , при условии, что после него идет Y

lookbehind - ретроспективная проверка - $(?<=Y)X$ - найти X , при условии, что до него идет Y

parameters: \$1 = 'Снам Flash calls', \$2 = 'Снам Flash calls', \$3 = '{cdr,me\$9rged}', \$4 = '2024-09-04 23:10:17'

`array_to_string(REGEXP_MATCHES(parameters, '(?<=\=)(.*)?(?=(,\\ \$\d+|$))', 'g'), 'g'), ';') AS arg_val)`

lookahead - опережающая проверка - `X(?:=Y)` - найти X, при условии, что после него идет Y

lookbehind - ретроспективная проверка - `(?:<=Y)X` - найти X, при условии, что до него идет Y

`*?` — ленивый режим.

parameters: \$1 = '**Снам Flash calls**', \$2 = '**Снам Flash calls**', \$3 = '**{cdr,me\$9rged}**', \$4 = '**2024-09-04 23:10:17**'

Заметочки:



Источники новых знаний:

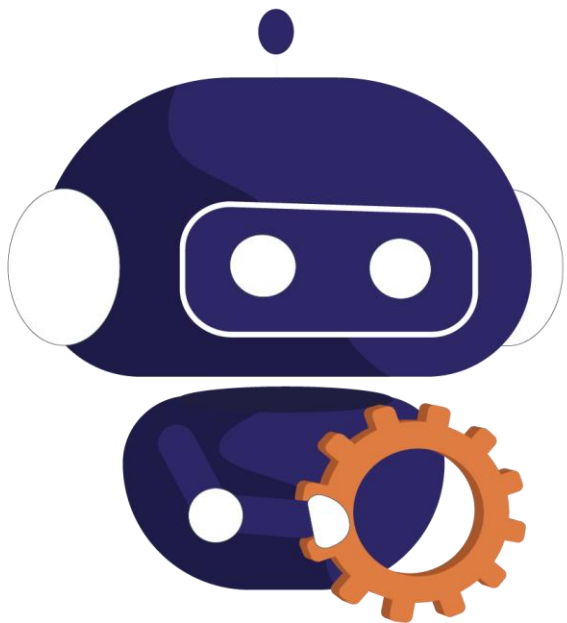
- Документация
- Группы в Telegram
- RSS-подписки на сайты
- Книги



Источники новых знаний:

- Документация
- Группы в Telegram
- RSS-подписки на сайты
- Книги
- **Конференции**





• Вопросы?



Спасибо за внимание!

Александр Никитин

Ведущий администратор БД PGMechanix

tg: <https://t.me/anikitindba>

<https://pgmech.ru/>

