



**ZABBIX** 5.0

МИГРАЦИЯ С MYSQL НА  
POSTGRESQL



**Александр Петров-Гаврилов**  
Инженер технической поддержки

# 01



## ЗАЧЕМ МИГРИРОВАТЬ?

- ☑ Вы лучше разбираетесь в PostgreSQL
- ☑ Вы захотели попробовать TimescaleDB
- ☑ Документация на русском языке

**ЕСТЬ ЛИ РАЗНИЦА?**





# ЕСТЬ ЛИ РАЗНИЦА?

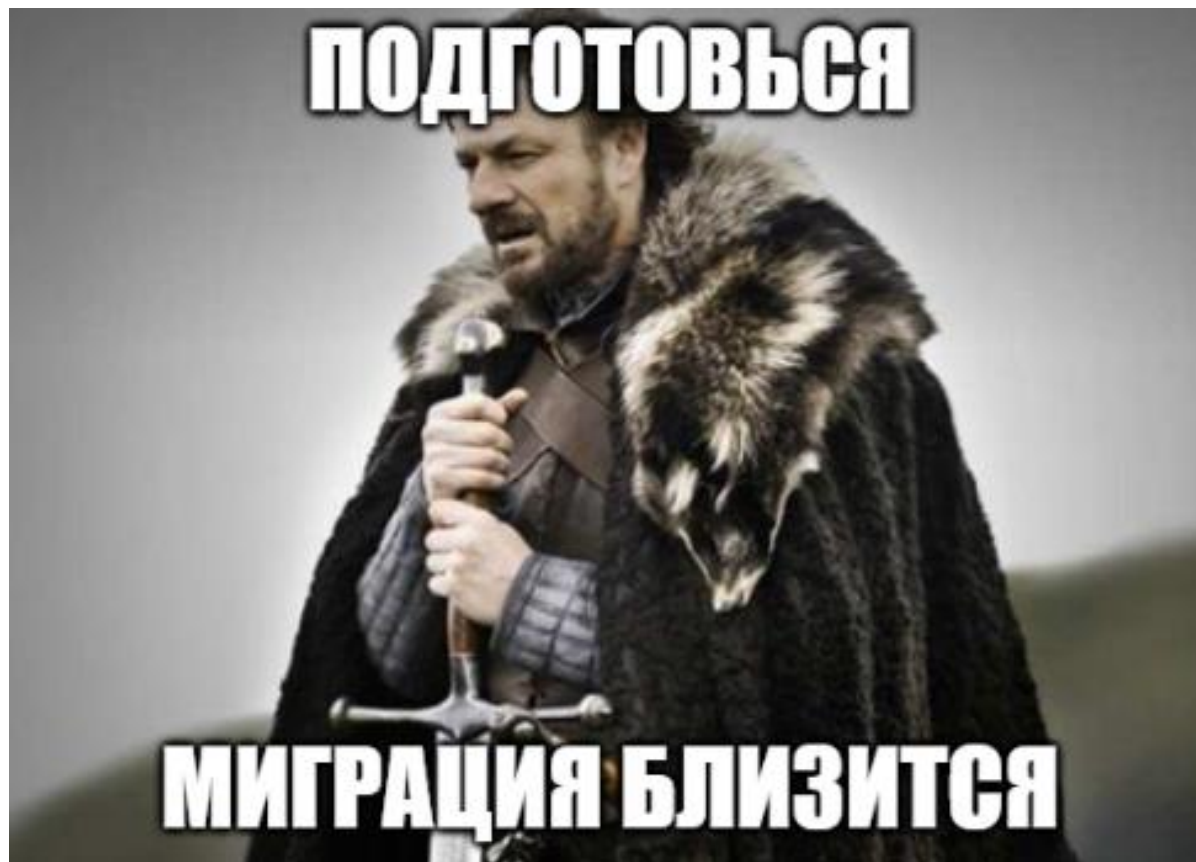
- ☑ Производительность PostgreSQL и MySQL практически одинакова для Zabbix DB.
- ☑ Тюнинг необходим(buffer и transaction log) и важен.
- ☑ Партиционирование для MySQL's немного проще чем PostgreSQL.
- ☑ Партиционирование PostgreSQL's поддерживает foreign key
- ☑ PostgreSQL бывает стабильнее в случае высокого IO

# 02

ПОДГОТОВКА



# КАК МИГРИРОВАТЬ?



# SETUP

- ✓ Zabbix 5.0
- ✓ Centos 7
- ✓ MariaDB 5.5.65

# ЧТО ПОНАДОБИТСЯ?

- ✓ PostgreSQL
- ✓ PGloader
- ✓ Исходный код Zabbix



# 03

КАК МИГРИРОВАТЬ?



# КАК МИГРИРОВАТЬ?



# ПОДГОТОВКА

1. Добавляем репозиторий и устанавливаем PostgreSQL

```
# yum install https://download.postgresql.org/pub/repos/yum/reporpms/EL-7-x86\_64/pgdg-redhat-repo-latest.noarch.rpm  
  
# yum install postgresql12-server  
  
# /usr/pgsql-12/bin/postgresql-12-setup initdb  
  
# systemctl enable postgresql-12  
# systemctl start postgresql-12
```

2. Устанавливаем pgloader

```
# yum install pgloader  
  
# pgloader -V  
# pgloader version "3.6.2"
```

# ПОДГОТОВКА

1. Создаем директорию для работы

```
# mkdir myzabbix-pgzabbix  
# cd myzabbix-pgzabbix
```

2. Скачиваем исходный код Zabbix

```
# yum install wget  
# wget https://cdn.zabbix.com/zabbix/sources/stable/5.0/zabbix-5.0.1.tar.gz
```

3. Распаковываем

```
# tar -zxvf zabbix-5.0.1.tar.gz
```



# ПОДГОТОВКА

1. Разделяем schema.sql на create.sql и alter.sql

```
# cd myzabbix-pgzabbix/zabbix-5.0.1/database/postgresql/  
  
# sed -n '/CREATE.*$/,/INSERT.*$/p' schema.sql | head -n-1 > create.sql  
  
# grep ALTER schema.sql > alter.sql
```

2. Создаём пользователя и базу данных в PostgreSQL

```
# sudo -u postgres createuser --pwprompt zabbix  
  
# sudo -u postgres createdb -O zabbix zabbix
```

# ПОДГОТОВКА

1. В той же директории где находятся create.sql и alter.sql создаём скрипт zabbix\_migrate.load

```
LOAD DATABASE
FROM mysql://zabbix:zabbix-password@localhost/zabbix
INTO postgresql://zabbix:zabbix-password@localhost/zabbix
WITH include no drop,
    truncate,
    create no tables,
    create no indexes,
    no foreign keys,
    reset sequences,
    data only
SET maintenance_work_mem TO '1024MB', work_mem to '256MB'
ALTER SCHEMA 'zabbix' RENAME TO 'public'
BEFORE LOAD EXECUTE create.sql
AFTER LOAD EXECUTE alter.sql;
```

# МИГРАЦИЯ

## 1. Останавливаем Zabbix сервер

```
systemctl stop zabbix-server
```

## 2. Запускаем pgloader

```
# pgloader zabbix-migrate.load
```

## 3. Видим warnings, но это нормально

```
2020-06-17T22:23:12.726000+01:00 WARNING Source column "public"."widget_field".
c"."widget_field"."type".
2020-06-17T22:23:12.726000+01:00 WARNING Source column "public"."widget_field".
public"."widget_field"."value_int".
2020-06-17T22:23:12.726000+01:00 WARNING Source column "public"."widget_field".
mn "public"."widget_field"."value_groupid".
```

## 4. Получаем результат

```
-----
COPY Threads Completion          0          4          11.983s
  Reset Sequences                0          3           0.051s
  Install Comments                0          0           0.000s
  after load                      0        223           2.424s
-----
Total import time                ✓    557393    22.2 MB    14.458s
```

# ПЕРЕПРОВЕРЯЕМ

```
maintenance_from | integer | | not null | 0 | plain |
ipmi_errors_from | integer | | not null | 0 | plain |
snmp_errors_from | integer | | not null | 0 | plain |
ipmi_error | character varying(2048) | | not null | ''::character varying | extended |
snmp_error | character varying(2048) | | not null | ''::character varying | extended |
jmx_disable_until | integer | | not null | 0 | plain |
jmx_available | integer | | not null | 0 | plain |
jmx_errors_from | integer | | not null | 0 | plain |
jmx_error | character varying(2048) | | not null | ''::character varying | extended |
name | character varying(128) | | not null | ''::character varying | extended |
flags | integer | | not null | 0 | plain |
templateid | bigint | | | | plain |
description | text | | not null | ''::text | extended |
tls_connect | integer | | not null | 1 | plain |
tls_accept | integer | | not null | 1 | plain |
tls_issuer | character varying(1024) | | not null | ''::character varying | extended |
tls_subject | character varying(1024) | | not null | ''::character varying | extended |
tls_psk_identity | character varying(128) | | not null | ''::character varying | extended |
tls_psk | character varying(512) | | not null | ''::character varying | extended |
proxy_address | character varying(255) | | not null | ''::character varying | extended |
auto_compress | integer | | not null | 1 | plain |
discover | integer | | not null | 0 | plain |
indexes:
"hosts_pkey" PRIMARY KEY, btree (hostid)
"hosts_1" btree (host)
"hosts_2" btree (status)
"hosts_3" btree (proxy_hostid)
"hosts_4" btree (name)
"hosts_5" btree (maintenanceid)
foreign-key constraints:
"c_hosts_1" FOREIGN KEY (proxy_hostid) REFERENCES hosts(hostid)
"c_hosts_2" FOREIGN KEY (maintenanceid) REFERENCES maintenances(maintenanceid)
"c_hosts_3" FOREIGN KEY (templateid) REFERENCES hosts(hostid) ON DELETE CASCADE
Referenced by:
TABLE "applications" CONSTRAINT "c_applications_1" FOREIGN KEY (hostid) REFERENCES hosts(hostid) ON DELETE CASCADE
TABLE "autoreg_host" CONSTRAINT "c_autoreg_host_1" FOREIGN KEY (proxy_hostid) REFERENCES hosts(hostid) ON DELETE CASCADE
TABLE "drules" CONSTRAINT "c_drules_1" FOREIGN KEY (proxy_hostid) REFERENCES hosts(hostid)
TABLE "group_prototype" CONSTRAINT "c_group_prototype_1" FOREIGN KEY (hostid) REFERENCES hosts(hostid) ON DELETE CASCADE
TABLE "host_discovery" CONSTRAINT "c_host_discovery_1" FOREIGN KEY (hostid) REFERENCES hosts(hostid) ON DELETE CASCADE
TABLE "host_discovery" CONSTRAINT "c_host_discovery_2" FOREIGN KEY (parent_hostid) REFERENCES hosts(hostid)
TABLE "host_inventory" CONSTRAINT "c_host_inventory_1" FOREIGN KEY (hostid) REFERENCES hosts(hostid) ON DELETE CASCADE
TABLE "host_tag" CONSTRAINT "c_host_tag_1" FOREIGN KEY (hostid) REFERENCES hosts(hostid) ON DELETE CASCADE
TABLE "hostmacro" CONSTRAINT "c_hostmacro_1" FOREIGN KEY (hostid) REFERENCES hosts(hostid) ON DELETE CASCADE
TABLE "hosts" CONSTRAINT "c_hosts_1" FOREIGN KEY (proxy_hostid) REFERENCES hosts(hostid)
TABLE "hosts" CONSTRAINT "c_hosts_3" FOREIGN KEY (templateid) REFERENCES hosts(hostid) ON DELETE CASCADE
TABLE "hosts_groups" CONSTRAINT "c_hosts_groups_1" FOREIGN KEY (hostid) REFERENCES hosts(hostid) ON DELETE CASCADE
TABLE "hosts_templates" CONSTRAINT "c_hosts_templates_1" FOREIGN KEY (hostid) REFERENCES hosts(hostid) ON DELETE CASCADE
TABLE "hosts_templates" CONSTRAINT "c_hosts_templates_2" FOREIGN KEY (templateid) REFERENCES hosts(hostid) ON DELETE CASCADE
TABLE "httpstest" CONSTRAINT "c_httpstest_2" FOREIGN KEY (hostid) REFERENCES hosts(hostid) ON DELETE CASCADE
TABLE "interface" CONSTRAINT "c_interface_1" FOREIGN KEY (hostid) REFERENCES hosts(hostid) ON DELETE CASCADE
TABLE "items" CONSTRAINT "c_items_1" FOREIGN KEY (hostid) REFERENCES hosts(hostid) ON DELETE CASCADE
TABLE "lld_override_optemplate" CONSTRAINT "c_lld_override_optemplate_2" FOREIGN KEY (templateid) REFERENCES hosts(hostid)
TABLE "maintenances_hosts" CONSTRAINT "c_maintenances_hosts_2" FOREIGN KEY (hostid) REFERENCES hosts(hostid) ON DELETE CASCADE
TABLE "opcommand_hst" CONSTRAINT "c_opcommand_hst_2" FOREIGN KEY (hostid) REFERENCES hosts(hostid)
TABLE "optemplate" CONSTRAINT "c_optemplate_2" FOREIGN KEY (templateid) REFERENCES hosts(hostid)
TABLE "screens" CONSTRAINT "c_screens_1" FOREIGN KEY (templateid) REFERENCES hosts(hostid) ON DELETE CASCADE
TABLE "task" CONSTRAINT "c_task_1" FOREIGN KEY (proxy_hostid) REFERENCES hosts(hostid) ON DELETE CASCADE
TABLE "widget_field" CONSTRAINT "c_widget_field_3" FOREIGN KEY (value_hostid) REFERENCES hosts(hostid) ON DELETE CASCADE
```



# ЕЩЁ РАЗ ПЕРЕПРОВЕРЯЕМ

23269	1592227969	0	136380984
31129	1592227969	12.4298095703125	136431172
23270	1592227970	0	137239896
31130	1592227970	0.36117265235391677	137290370
31011	1592227971	0	138603635
25371	1592227971	0	138616784
31012	1592227972	0	140274435
31072	1592227972	30.542634000000007	140631993
31013	1592227973	0	141245449
10073	1592227973	1.1819641414622033	141264065
23273	1592227973	12.431907653808594	141274420
31014	1592227974	0	141880662
10074	1592227974	0	141899771
23274	1592227974	0	141913658
31135	1592227975	0	142733002
10075	1592227975	0	142766133
23275	1592227975	0.2216546269926503	142793306
31136	1592227976	0.2216546269926503	143406494
10076	1592227976	0.4661249638671019	143450775
23276	1592227976	0.24871826171875	143494775
31137	1592227977	0.25081634521484375	144238188
10077	1592227977	0.04994163240764782	144268763
23277	1592227977	1.764600899390716	144333164
31018	1592227978	12.43600845336914	145288880
10078	1592227978	0.16647062165123577	145311145
31019	1592227979	0.39054713794277507	145765902
29162	1592227982	0.268276	149146486
29823	1592227983	0	149259219
29163	1592227983	354.328873496089	149528576
31024	1592227984	0	150226345
29164	1592227984	0	150618585
31025	1592227985	0.2216546269926503	151480705
29165	1592227985	0	151604853
31026	1592227986	0.2655029296875	152556662
29166	1592227986	0.033518	152858985
29167	1592227987	0	154439526
29168	1592227988	0	155982606
29169	1592227989	0	157480167

# ФИНАЛЬНЫЕ ШТРИХИ

1. Удаляем Zabbix сервер для работы с MySQL

```
# yum remove Zabbix-server-mysql
```

2. Удаляем веб интерфейс для работы с MySQL

```
# yum remove zabbix-web-*
```

3. Устанавливаем Zabbix сервер для работы с PostgreSQL

```
# yum install zabbix-server-pgsql
```

4. Устанавливаем веб интерфейс для работы с PostgreSQL

```
# yum install zabbix-web-pgsql-scl zabbix-apache-conf-scl
```

# ФИНАЛЬНЫЕ ШТРИХИ

1. Редактируем новый zabbix\_server.conf

```
# vi /etc/zabbix/zabbix_server.conf
```

2. Добавляем наш пароль

```
# DBPassword=zabbix
```

3. Удаляем предыдущую конфигурацию веб интерфейса

```
# rm /etc/zabbix/web/zabbix.conf.php
```

4. Снова раскомментируйте строку и укажите свой часовой пояс

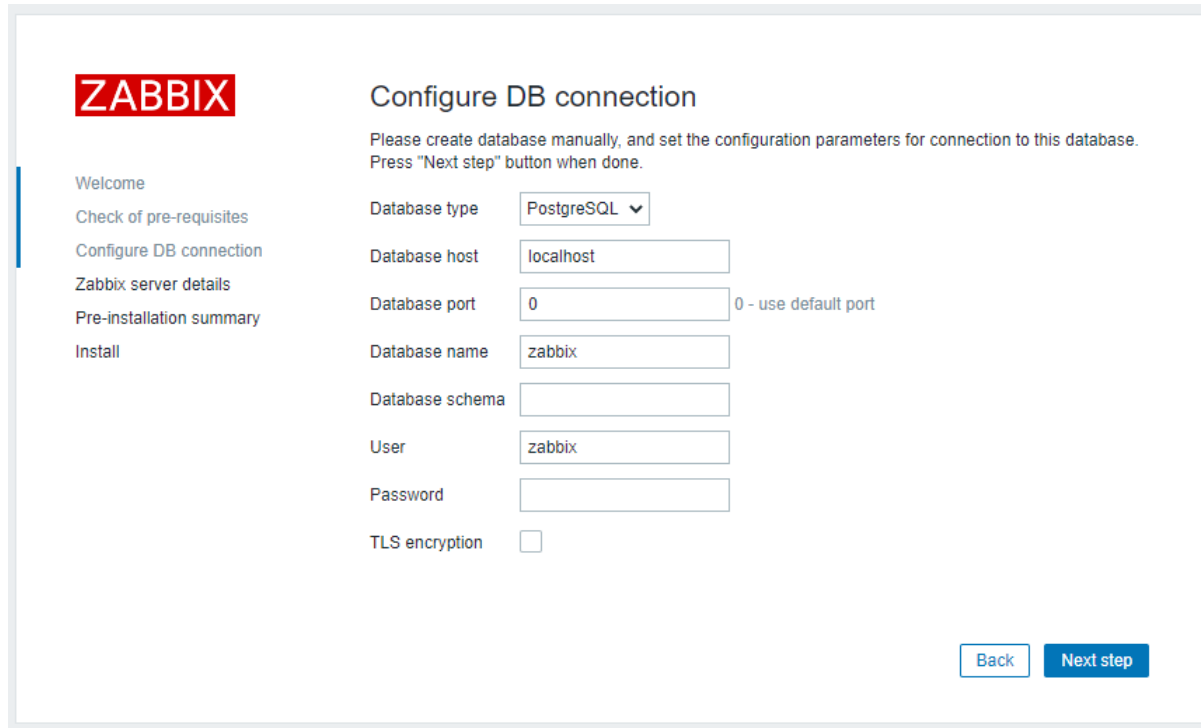
```
# vi /etc/httpd/conf.d/zabbix.conf
```

# ФИНАЛЬНЫЕ ШТРИХИ

1. Запустите Zabbix сервер, перезагрузите httpd

```
# systemctl restart zabbix-server httpd
```

2. Снова настройте веб интерфейс, но уже для PostgreSQL



**ZABBIX**

Welcome  
Check of pre-requisites  
Configure DB connection  
Zabbix server details  
Pre-installation summary  
Install

### Configure DB connection

Please create database manually, and set the configuration parameters for connection to this database.  
Press "Next step" button when done.

Database type: PostgreSQL ▼

Database host: localhost

Database port: 0 0 - use default port

Database name: zabbix

Database schema:

User: zabbix

Password:

TLS encryption:

Back Next step



# И СНОВА ПЕРЕПРОВЕРЯЕМ

The screenshot displays the Zabbix web interface. The left sidebar contains navigation menus for Monitoring, Inventory, Reports, Configuration, and Administration. The main content area is titled "Global view" and shows system information and a table of problems.

**ZABBIX** postgresql

Global view

All dashboards / Global view

### System information

Parameter	Value	Details
Zabbix server is running	Yes	localhost:10051
Number of hosts (enabled/disabled/templates)	148	3 / 0 / 145
Number of items (enabled/disabled/not supported)	444	404 / 0 / 40
Number of triggers (enabled/disabled [problem/ok])	288	288 / 0 [0 / 288]
Number of users (online)	2	1
Required server performance, new values per second	6.27	

### Problems

Time	Info	Host	Problem • Severity
------	------	------	--------------------

# ПОСЛЕДНИЙ ШАГ

1. Остановите MySQL

```
# systemctl stop mariadb
```

# ПОЗДРАВЛЯЮ, **МЫ МИГРИРОВАЛИ**



Спасибо!

