initMAX

Webinar

Zabbix - Migration from MySQL to PostgreSQL

all our microphones are muted

ask your questions in Q&A, not in the Chat

use Chat for discussion, networking or applause

1

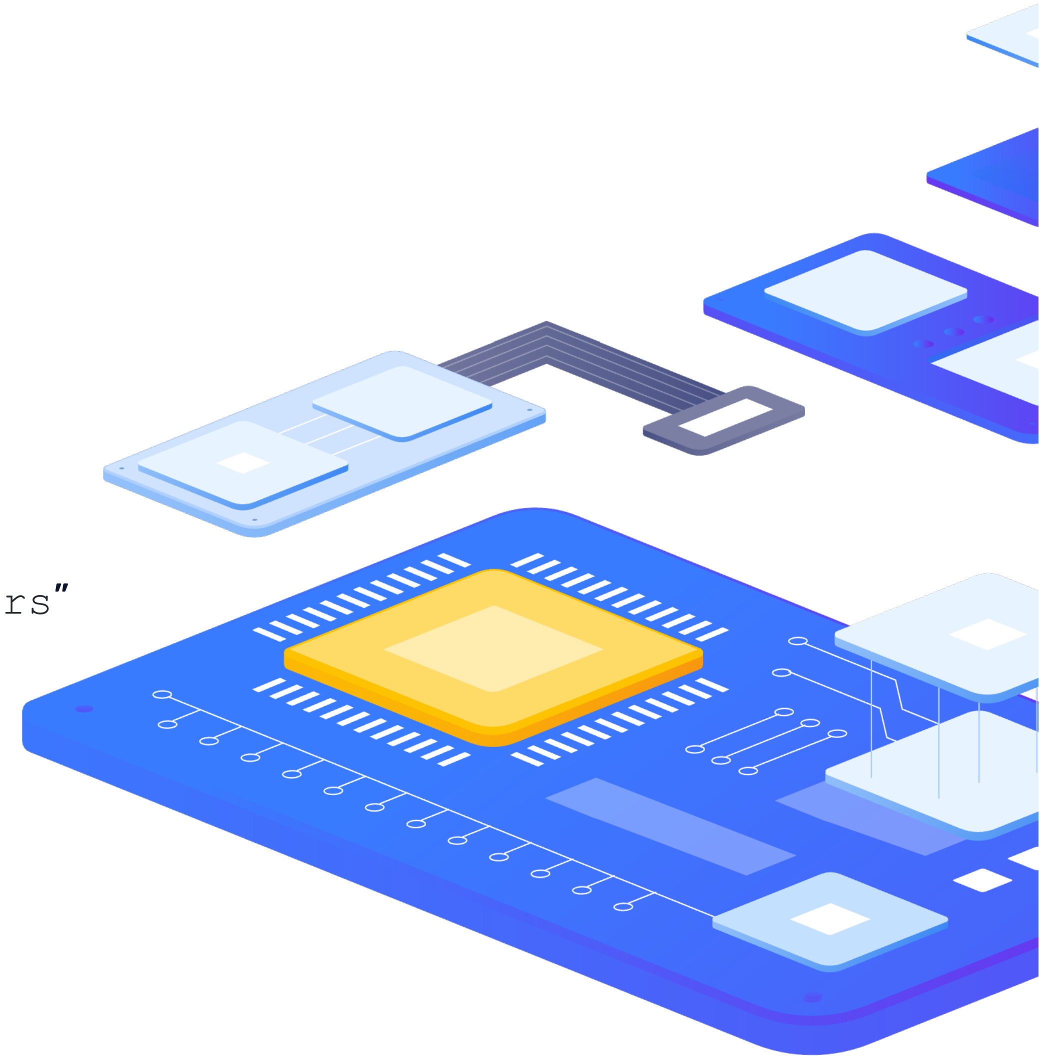
Why migrate?

Zabbix - Migration from MySQL to PostgreSQL

Why migrate?

What is a main reason for migration?

- Better community support
- Better performance and stability
- Zabbix history syncer is not ready for master-master replication
- Out of the box **partitioning and compression** via TimescaleDB
- Easy and strong grow for HA
- You don’t need to care about “`log_bin_trust_function_creators`”
- Many more…



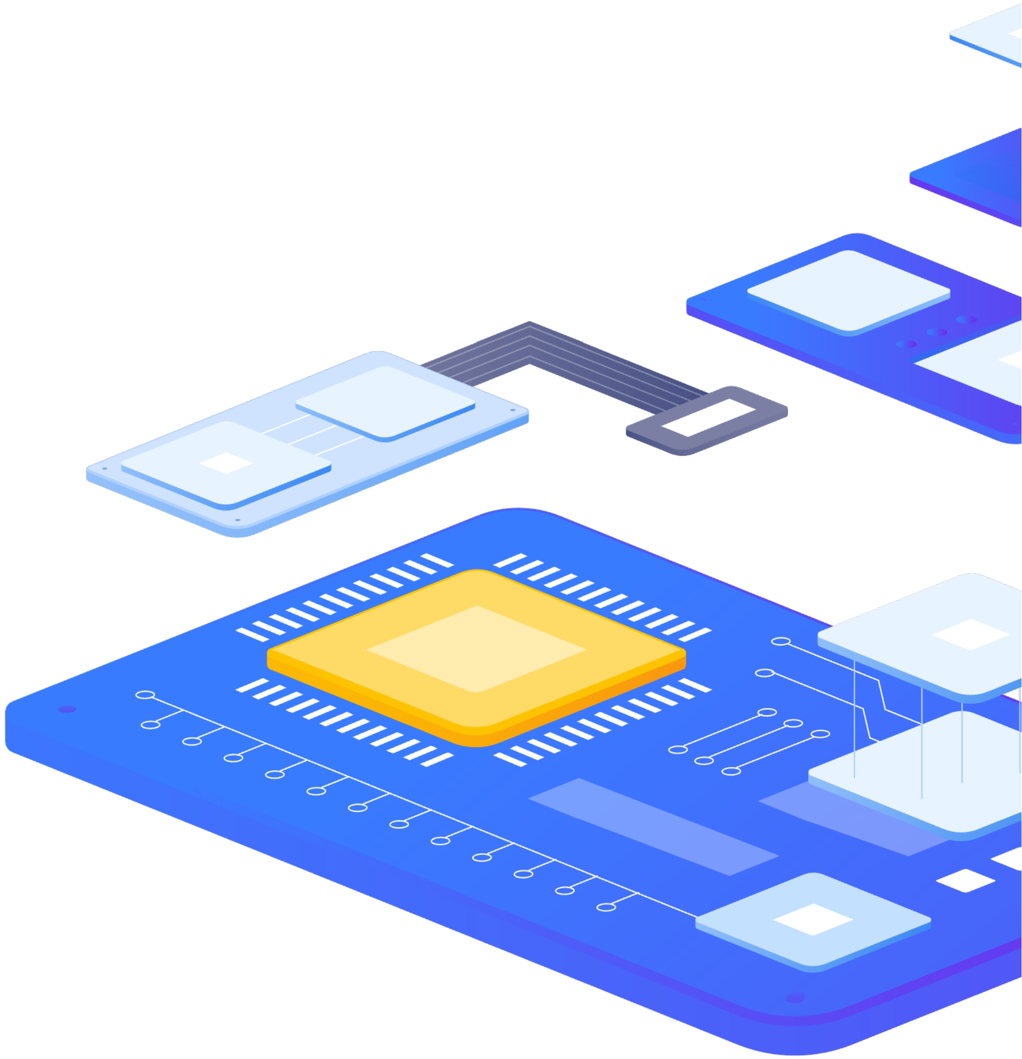
2

Important notice

Important notice

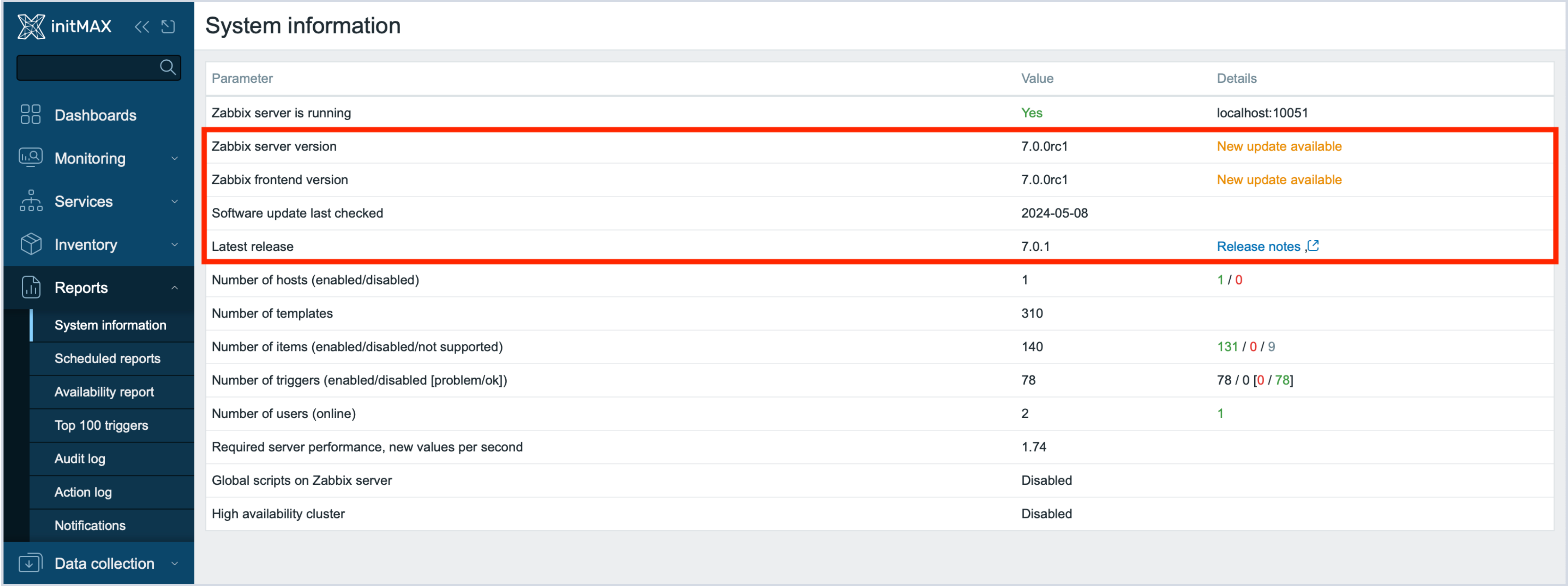
Important notice

- **We warn you in advance that you do the migration at your own risk, and we bear no responsibility for any damage caused by unprofessional intervention.**
- **Take care about Zabbix schema version!**
- **Don’t skip any steps!**
- **Be careful with DB triggers!**
- If you need help, we are ready to help you with our team of Zabbix Certified Experts and also our team of Postgres Certified Engineers.
- Make sure your Zabbix environment is in good health, and you don't have any problems with your MySQL database, including all customizations.
- Check for supported version by your Zabbix Server
- Check free disk space on DB server



Important notice

- Check all potential issues in log and on Zabbix > Reports > **System information** page before migration!



System information

Parameter	Value	Details
Zabbix server is running	Yes	localhost:10051
Zabbix server version	7.0.0rc1	New update available
Zabbix frontend version	7.0.0rc1	New update available
Software update last checked	2024-05-08	
Latest release	7.0.1	Release notes
Number of hosts (enabled/disabled)	1	1 / 0
Number of templates	310	
Number of items (enabled/disabled/not supported)	140	131 / 0 / 9
Number of triggers (enabled/disabled [problem/ok])	78	78 / 0 [0 / 78]
Number of users (online)	2	1
Required server performance, new values per second	1.74	
Global scripts on Zabbix server	Disabled	
High availability cluster	Disabled	

3

Preparation and rules

Zabbix - Migration from MySQL to PostgreSQL

Preparation and rules

First steps and checks

- **Check your running Zabbix Server version (be sure your Zabbix is not pending for restart)**

```
› zabbix_server -V
```

- Output:

```
zabbix_server (Zabbix) 7.0.6
Revision c1d7a081969 20 November 2024, compilation time: Nov 20 2024 00:00:00
```

- This is important for future steps if your version is pending for restart after update you don’t have maybe applied some DB patches.

Preparation and rules

First steps and checks

- **Check DB triggers**

```
› mysql
› use zabbix;
› SHOW TRIGGERS\G
```

- **Result 1/2**

```
Empty set (0.00 sec)
```

- **In these conditions, you can skip section regarding to triggers**

Preparation and rules

First steps and checks

- **Result 2/2 (you need to care about triggers in your migration)**

```
*************************** 1. row ***************************
             Trigger: hosts_name_upper_insert
               Event: INSERT
               Table: hosts
           Statement: set new.name_upper=upper(new.name)
              Timing: BEFORE
             Created: 2024-02-27 09:59:58.09
            sql_mode:
ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO
,NO_ENGINE_SUBSTITUTION
             Definer: zabbix@localhost
character_set_client: utf8mb4
collation_connection: utf8mb4_0900_ai_ci
  Database Collation: utf8mb4_bin
```

Installing dependencies

Repository, DB and pgloader

- **First, we add the official PostgreSQL repository that we recommend for installation.**

```
› yum install https://download.postgresql.org/pub/repos/yum/reporpms/EL-9-x86_64/pgdg-redhat-repo-latest.noarch.rpm
```

- **Installing of PostgreSQL server and pgloader**

```
› yum install postgresql17-server
› /usr/pgsql-17/bin/postgresql-17-setup initdb
› systemctl enable postgresql-17
› systemctl start postgresql-17
› yum install pgloader
```

Zabbix - Migration from MySQL to PostgreSQL

Preparing for migration

Temp directory

- **For ease of use in next step we create tmp directory. Make sure you have enough free space in this location too.**

```
› mkdir /tmp/zabbix-db-migration/ && cd $_
```

Schema parsing

Download schema

- **We need to use EXACTLY same version of source codes like our Zabbix version!**

```
wget https://cdn.zabbix.com/zabbix/sources/development/7.0/zabbix-7.0.0.tar.gz
```

- Unpack source code

```
tar -zxvf zabbix-7.0.0.tar.gz
```

- Open directory with schema.sql file

```
cd /tmp/zabbix-db-migration/zabbix-7.0.0/database/postgresql/
```

Schema parsing

Parse schema

- 1/5 Parse only basic schema for creating tables

```
grep -v 'ALTER TABLE ONLY' schema.sql | grep -v INSERT | grep -v 'CREATE INDEX' | grep -v
'CREATE UNIQUE INDEX' > /tmp/zabbix-db-migration/create_tables.sql
```

- 2/5 Remove trigger functions from this file

```
sed -i '/create\ or\ replace\ function/,$d' /tmp/zabbix-db-migration/create_tables.sql
```

- 3/5 Create a separate file containing operations related to triggers (if applicable)

```
awk '/INSERT INTO dbversion/{p=1;next} /ALTER TABLE/{p=0} p' schema.sql > /tmp/zabbix-db-
migration/triggers.sql
```

Schema parsing

Parse schema

- 4/5 Select only index creation operations and store them in a separate file

```
grep -E 'CREATE INDEX|CREATE UNIQUE INDEX' schema.sql > /tmp/zabbix-db-migration/create_index.sql
```

- 5/5 Create a separate file containing ALTER TABLE ONLY operations

```
grep 'ALTER TABLE ONLY' schema.sql > /tmp/zabbix-db-migration/alter_table.sql
```

- Check if you have 4 sql files (if you don’t have triggers, you will only have 3 of them)

```
ls /tmp/zabbix-db-migration/*.sql
```

Preparing PostgreSQL

DB preparation

- **Go back to our temporary directory**

```
cd /tmp/zabbix-db-migration/
```

- **Create a database user for Zabbix, you will be prompted to enter a password**

```
sudo -u postgres createuser --pwprompt zabbix
```

- **Create Zabbix DB**

```
sudo -u postgres createdb -O zabbix zabbix
```

Zabbix - Migration from MySQL to PostgreSQL

Preparing PostgreSQL

DB preparation

- Finally create our stripped db schema

```
› sudo -u postgres psql --host=127.0.0.1 --dbname=zabbix --username=zabbix -f /tmp/zabbix-db-
  migration/create_tables.sql
```

- For compatibility with the `pgloader` utility, temporarily set the encryption hash to ‘md5’ and change the password of the created database user so that it is regenerated in the given hash algorithm. For simplicity, ideally use the same password as you’ve entered on creation of this user.

```
› sudo -u postgres psql -c "SET password_encryption='md5';"
› sudo -u postgres psql -c "ALTER ROLE zabbix WITH PASSWORD '***********';"
```

Preparing MySQL

DB preparation

- Force MySQL to use mysql_native_password

```
› nano /etc/my.cnf.d/mysql-server.cnf
```

- Change default authentication plugin in MySQL

```
[mysqld]
...
default-authentication-plugin=mysql_native_password
```

- Apply new settings by restarting MySQL server

```
› systemctl restart mysqld
```

Preparing MySQL

DB preparation

- **Update old Zabbix user with new password in the old format**

```
› mysql -e "ALTER USER 'zabbix'@'localhost' IDENTIFIED WITH mysql_native_password BY
  '*********';"
```

pgloader

Config preparation

- Create new file for `pgloader` configuration

```
› nano /tmp/zabbix-db-migration/pgloader.conf
```

pgloader

Config preparation

- Content of this new config file, don’t forget to **change passwords**

```
LOAD DATABASE
FROM mysql://zabbix:**********@127.0.0.1/zabbix
INTO postgresql://zabbix:**********@127.0.0.1/zabbix
WITH include no drop,
truncate,
create no tables,
create no indexes,
no foreign keys,
reset sequences,
data only,
prefetch rows = 100,
batch rows = 1000,
batch concurrency = 1
ALTER SCHEMA 'zabbix' RENAME TO 'public';
```

4

Migration

Migration

Turning off all services and start migration

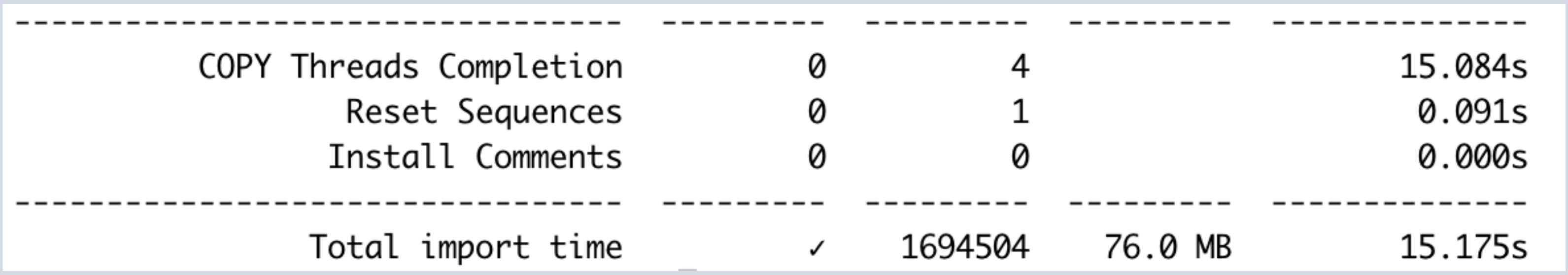
- We need to turn off all applications which are using Zabbix (Graphana also needs to be turned off)

```
› systemctl stop zabbix-server httpd
```

- Now is the time to start `pgloader` with its configuration

```
› pgloader --root-dir=/tmp/zabbix-db-migration/data /tmp/zabbix-db-migration/pgloader.conf
```

- Check for mark on the last line



```
---------------------------------  ---------  ---------  ---------  --------------
          COPY Threads Completion          0          4                  15.084s
                  Reset Sequences          0          1                   0.091s
                 Install Comments          0          0                   0.000s
---------------------------------  ---------  ---------  ---------  --------------
                Total import time          ✓    1694504    76.0 MB        15.175s
```

Migration

Turn back SCRAM-SHA-256 for your new Zabbix user

- `pgloader` already finished all its work, and we can now to turn back SCRAM security

```
› sudo -u postgres psql -c "SET password_encryption='SCRAM-SHA-256';"
› sudo -u postgres psql -c "ALTER ROLE zabbix WITH PASSWORD '************';"
```

Migration

Run all our sql scripts

- Create schema for indexes

```
sudo -u postgres psql --host=127.0.0.1 --dbname=zabbix --username=zabbix -f /tmp/zabbix-db-migration/create_index.sql
```

- Create schema for the alter table

```
sudo -u postgres psql --host=127.0.0.1 --dbname=zabbix --username=zabbix -f /tmp/zabbix-db-migration/alter_table.sql
```

- Create schema for triggers (if applicable)

```
sudo -u postgres psql --host=127.0.0.1 --dbname=zabbix --username=zabbix -f /tmp/zabbix-db-migration/triggers.sql
```

Migration

DB data has been migrated, now is the time for a cleanup

- Run vacuum

```
sudo -u postgres vacuumdb --dbname=zabbix --analyze --username=postgres --jobs=$(grep -c
processor /proc/cpuinfo)
```

- You can turn off your MySQL instance

```
systemctl stop mysqld
```

Zabbix - Migration from MySQL to PostgreSQL

Migration

We need to install packages for PostgreSQL support instead of MySQL

- Uninstall Zabbix MySQL packages

```
yum remove zabbix-server-mysql zabbix-web-mysql
```

- Install new packages with support of PostgreSQL

```
yum install zabbix-server-pgsql zabbix-web-pgsql zabbix-apache-conf
```

- Fix Zabbix server configuration (reinstallation removes your old config for Zabbix server, old config still exists and can be used after renaming, but in this example, we are using the new one)

```
nano /etc/zabbix/zabbix_server.conf
```

Migration

Change default Zabbix server config

- You need to change password to the database. Also, it's a good idea to modify other parameters to fit your environment

```
DBPassword=*******
```

- Start all services

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

- Check the log file

```
less /var/log/zabbix/zabbix_server.log
```

Migration - Frontend

Fix Zabbix frontend to work with PostgreSQL

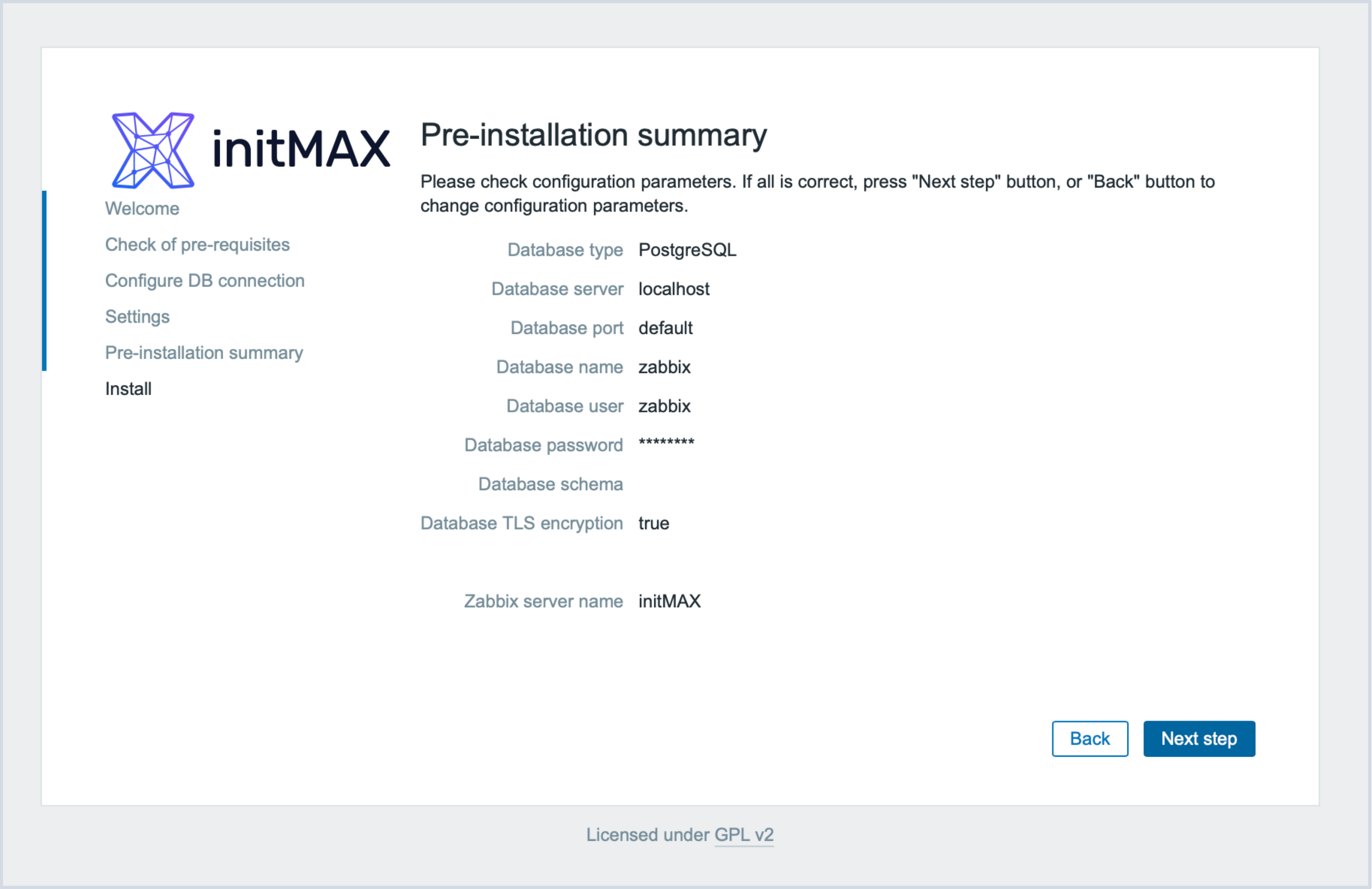
- You can change MYSQL in `/etc/zabbix/web/zabbix.conf.php` to POSTGRESQL or create s new config entirely. Our preference is to drop the old config and create new one.

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

- Continue with reinstallation in your browser default address is http://IP_OF_YOUR_ZABBIX_SERVER/zabbix

Zabbix - Migration from MySQL to PostgreSQL

Migration - Frontend



Pre-installation summary

Please check configuration parameters. If all is correct, press "Next step" button, or "Back" button to change configuration parameters.

Database type	PostgreSQL
Database server	localhost
Database port	default
Database name	zabbix
Database user	zabbix
Database password	********
Database schema	
Database TLS encryption	true
Zabbix server name	initMAX

Licensed under GPL v2

5

Post migration steps

Post migration steps

Now it's time to install TimescaleDB

- Let’s start by adding the official repository (don’t use any other repository!)

```
tee /etc/yum.repos.d/timescale_timescaledb.repo <<EOL
[timescale_timescaledb]
name=timescale_timescaledb
baseurl=https://packagecloud.io/timescale/timescaledb/el/$(rpm -E %{rhel})/\$basearch
repo_gpgcheck=1
gpgcheck=0
enabled=1
gpgkey=https://packagecloud.io/timescale/timescaledb/gpgkey
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt
metadata_expire=300
EOL
```

Zabbix - Migration from MySQL to PostgreSQL

Post migration steps

Installation of required packages

- For TimesacleDB you basically need only 2 packages

```
yum install timescaledb-2-postgresql-17 timescaledb-2-loader-postgresql-17
```

- Now run `timescaledb-tune`, this small script tests your configuration and helps you with enabling timescaledb extension. This script also tunes your PostgreSQL for usage with timescaledb. Don’t forget to change the value of `max-conns` parameter to fit your environment. The answer to all the questions is basically YES.

```
timescaledb-tune --pg-config /usr/pgsql-17/bin --max-conns=125
```

Post migration steps

Enable TimescaleDB for your Zabbix database

- Turn off Zabbix server first and restart PostgreSQL to apply the new configuration

```
systemctl stop zabbix-server
systemctl restart postgresql-17
```

- Activate TimescaleDB for the Zabbix database

```
echo "CREATE EXTENSION IF NOT EXISTS timescaledb CASCADE;" | sudo -u postgres psql --dbname=zabbix
```

Zabbix - Migration from MySQL to PostgreSQL

Post migration steps

Enable TimescaleDB for your Zabbix database

- Start migration to chunks
- This operation can take some time, depending on your history and trend data (starting with Zabbix 7.0 also audit table is migrated to chunks)

```
› sudo -u postgres psql --host=127.0.0.1 --dbname=zabbix --username=zabbix -f /tmp/zabbix-db-migration/zabbix-7.0.0/database/postgresql/timescaledb/schema.sql
```

- After successfully enabling TimescaleDB you can start your Zabbix server again

```
› systemctl start zabbix-server
```

Post migration steps

Some additional tips and tricks

- Don’t forget to delete MySQL completely after some period of time
- Don’t forget to set up PostgreSQL monitoring
- Don’t forget to configure backup (`pgBackRest`, `pgdump`,..)
- Tune DB for your specific environment
- In case of HA, best practice now is using a `Patroni cluster` (We have official Certified training for this product)
- After migration in large environments, you can encounter some awkwardness in your monitoring. This is a side effect of your new database performance. Usually, some parts of Zabbix are working better now and you have freed additional CPU resources.
- Don’t skip any steps especially alter tables or triggers, this topic is really important!
- If you need to speed up your frontend you can use `pgbouncer`.

- You can ask us for help, our specialists are ready to assist you.

Zabbix - Migration from MySQL to PostgreSQL

Turn key solution from initMAX

We are certified Zabbix Premium Partner and Cybertec Certified Partner

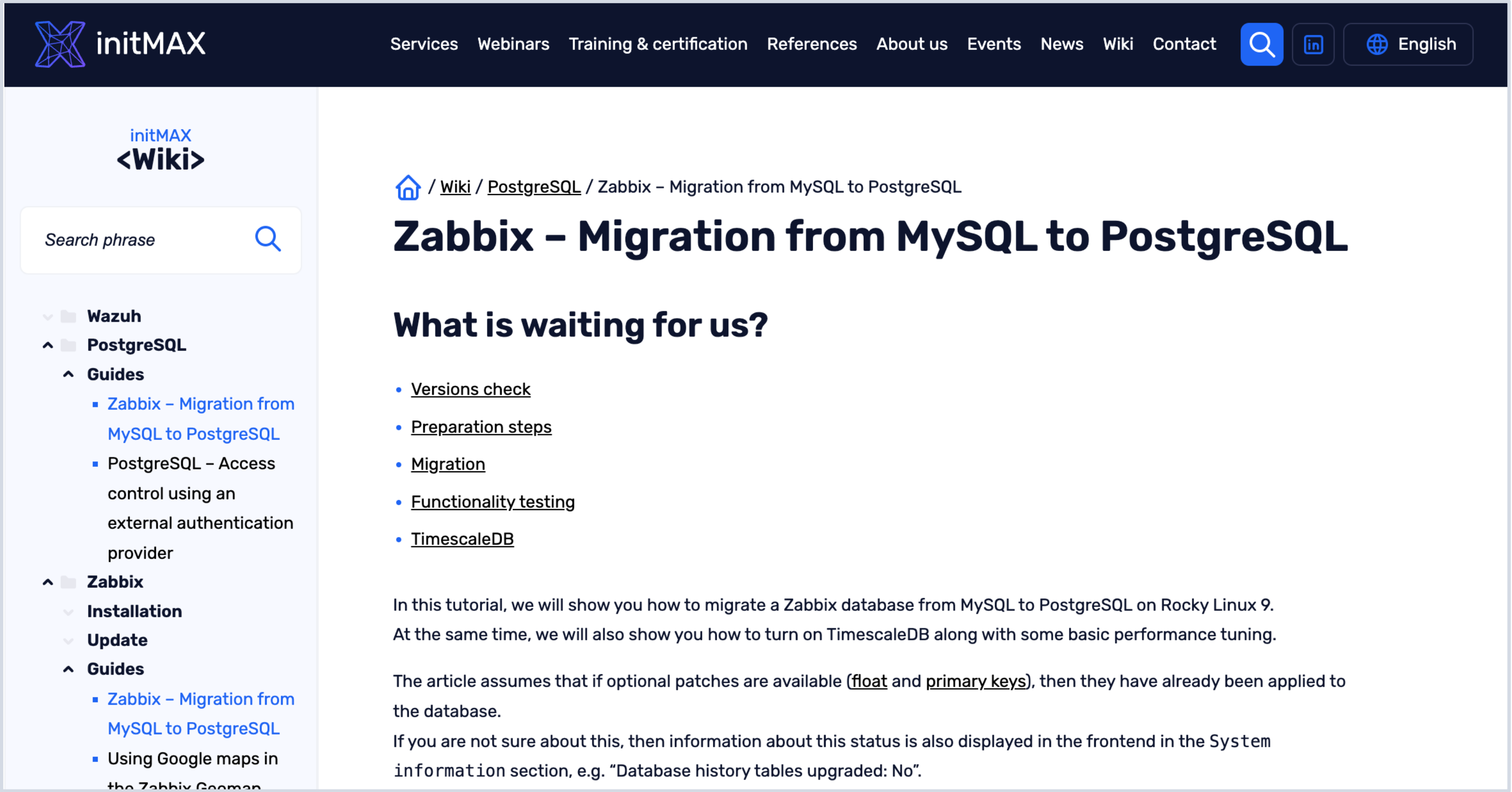
- We can help you with all your topics including knowledge transfer
- Some operations cannot be reverted when migration gone wrong
- https://www.initmax.com/contact/ EN
- https://www.initmax.cz/kontakt/ CZ

ZABBIX
PREMIUM PARTNER

CYBERTEC
POSTGRESQL SERVICES & SUPPORT
OFFICIAL PARTNER

Zabbix - Migration from MySQL to PostgreSQL

You can use our updated Wiki


initMAX
Services
Webinars
Training & certification
References
About us
Events
News
Wiki
Contact
English
initMAX
<Wiki>
Search phrase
Wazuh
PostgreSQL
Guides
Zabbix – Migration from MySQL to PostgreSQL
PostgreSQL – Access control using an external authentication provider
Zabbix
Installation
Update
Guides
Zabbix – Migration from MySQL to PostgreSQL
Using Google maps in the Zabbix Geomap

/ Wiki / PostgreSQL / Zabbix – Migration from MySQL to PostgreSQL

Zabbix – Migration from MySQL to PostgreSQL

What is waiting for us?

- Versions check
- Preparation steps
- Migration
- Functionality testing
- TimescaleDB

In this tutorial, we will show you how to migrate a Zabbix database from MySQL to PostgreSQL on Rocky Linux 9.
At the same time, we will also show you how to turn on TimescaleDB along with some basic performance tuning.

The article assumes that if optional patches are available (float and primary keys), then they have already been applied to the database.
If you are not sure about this, then information about this status is also displayed in the frontend in the `System information` section, e.g. “Database history tables upgraded: No”.

https://www.initmax.com/wiki/ EN
https://www.initmax.cz/wiki/ CZ

6

Demonstration

initMAX

Questions?

Contact us:

Phone:	›	+420 800 244 442
Web:	›	https://www.initmax.cz
Email:	›	tomas.hermanek@initmax.cz
LinkedIn:	›	https://www.linkedin.com/company/initmax
Twitter:	›	https://twitter.com/initmax
Tomáš Heřmánek:	›	+420 732 447 184