



Webinar

Monitoring PostgreSQL with Zabbix

all our microphones are muted

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

use Chat for discussion, networking or applause



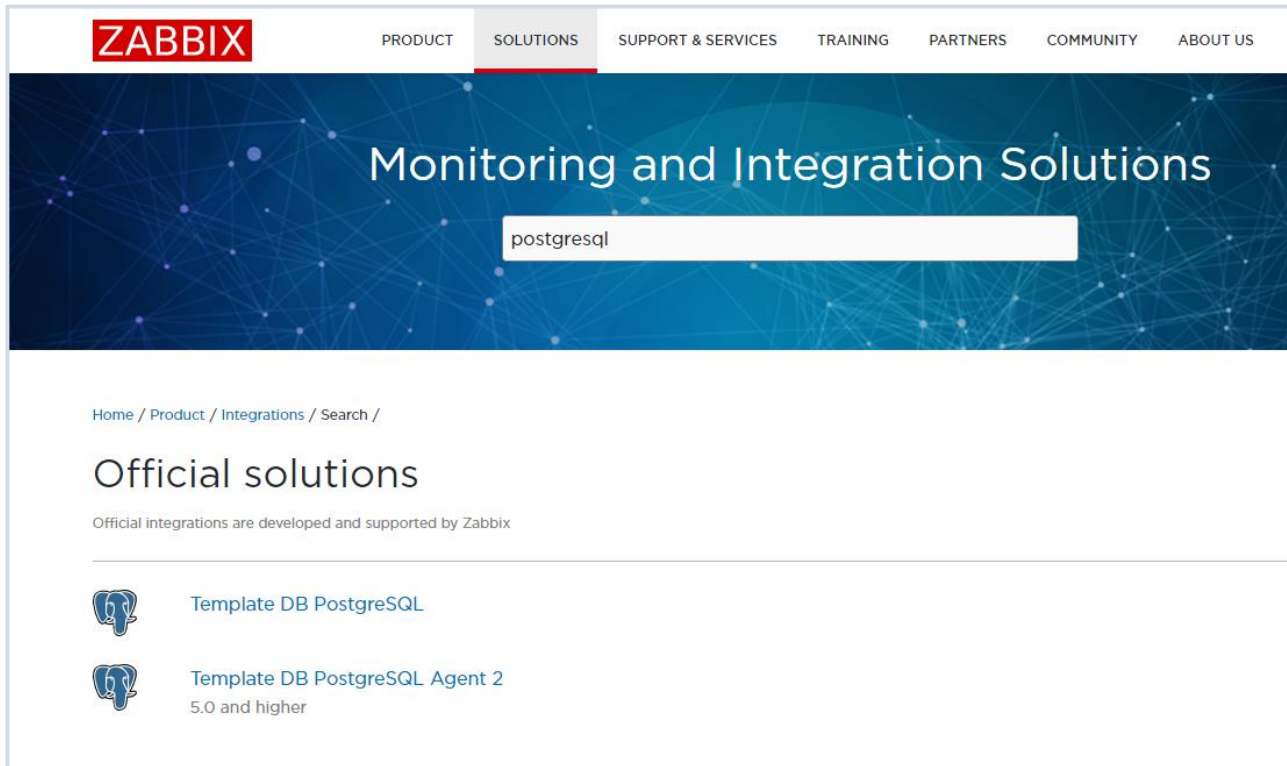
1

PostgreSQL connection variants



Monitoring PostgreSQL with Zabbix

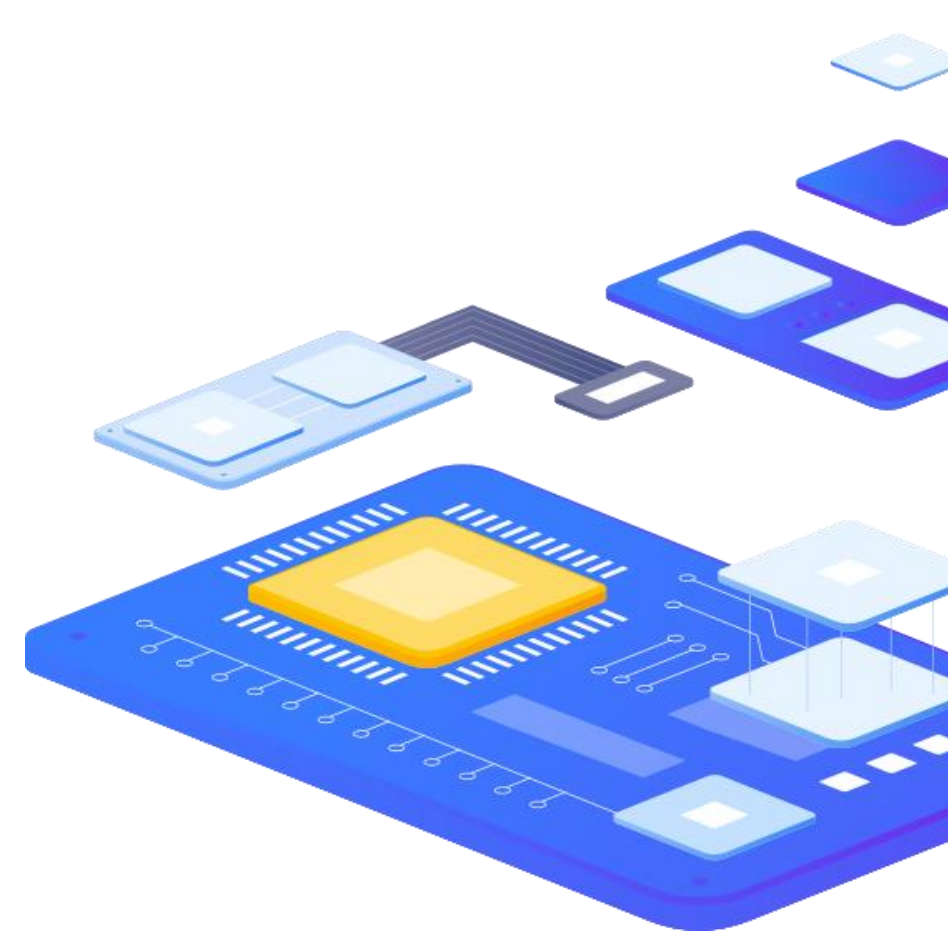
PostgreSQL connection from Zabbix

- › PostgreSQL by Zabbix agent
- › PostgreSQL by Zabbix agent 2
- › Database monitor - ODBC



The screenshot shows the Zabbix website's search results page. The navigation bar includes 'ZABBIX', 'PRODUCT', 'SOLUTIONS', 'SUPPORT & SERVICES', 'TRAINING', 'PARTNERS', 'COMMUNITY', and 'ABOUT US'. The main heading is 'Monitoring and Integration Solutions'. A search bar contains the text 'postgresql'. Below the search bar, the breadcrumb trail reads 'Home / Product / Integrations / Search /'. The section is titled 'Official solutions' with the subtext 'Official integrations are developed and supported by Zabbix'. Two results are listed:

-  Template DB PostgreSQL
-  Template DB PostgreSQL Agent 2
5.0 and higher

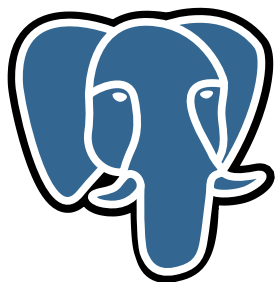


Monitoring PostgreSQL with Zabbix

Zabbix agent - User parameters

- ▶ Extend Zabbix agent with User Parameters
 - ▶ `template_db_postgresql.conf`
 - ▶ Sql scripts from `postgresql/` to Zabbix agent home directory `/var/lib/zabbix/`
- ▶ Create sql user, grant permissions
- ▶ Import and link `template_db_postgresql.yaml`
- ▶ Set `{$PG.HOST}`, `{$PG.PORT}`, `{$PG.USER}`, `{$PG.PASSWORD}` and `{$PG.DB}` macros values.

```
UserParameter=pgsql.dbstat[*], psql -qtAX postgresql://"${3}":"${4}"@"${1}":"${2}" cantainment/"${5}" -f  
"/var/lib/zabbix/postgresql/pgsql.dbstat.sql"
```



▶ Zabbix agent



Monitoring PostgreSQL with Zabbix

Zabbix agent2 PostgreSQL plugin

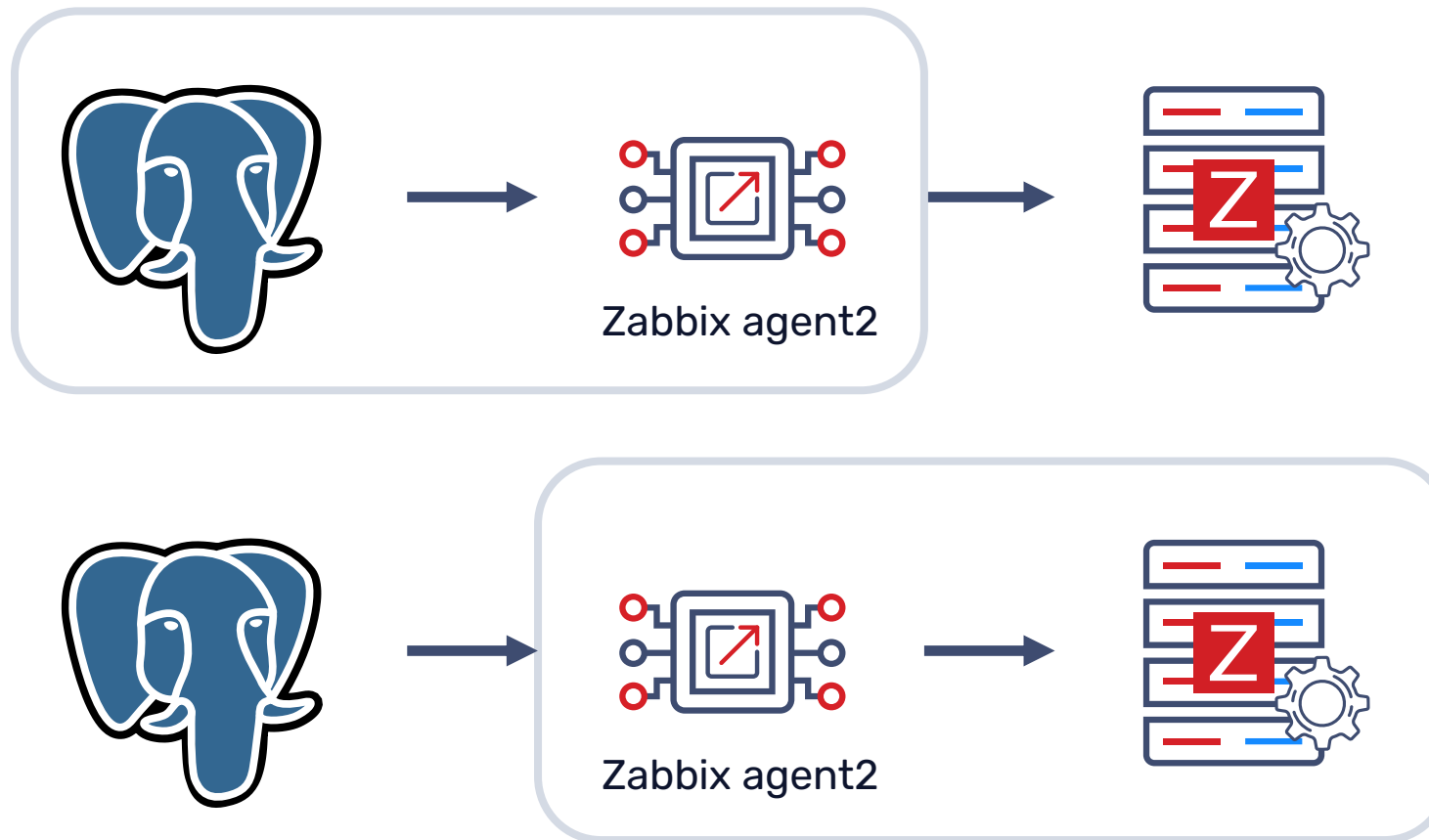
- › Install Zabbix agent2 PostgreSQL plugin
- › Create sql user, grant permissions
- › Import and link template_db_postgresql.yaml
- › Set in the `{$PG.URI}` macro the system data source name of the PostgreSQL instance such as `<protocol(host:port)>`.
- › Set the user name and password in host macros (`{$PG.USER}` and `{$PG.PASSWORD}`)



Monitoring PostgreSQL with Zabbix

Zabbix agent2 PostgreSQL plugin

- ▶ Location of Zabbix Agent, able to monitor several PostgreSQL instances by one Agent



Monitoring PostgreSQL with Zabbix

ODBC driver

- › Install ODBC driver

- › Debian: ODBC Driver for PostgreSQL is compatible with UnixODBC driver manager

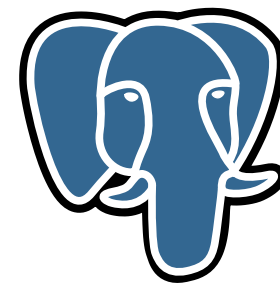
```
apt install odbcinst1debian2 libodbc1 odbcinst unixodbc
```

- › RedHat:

```
dnf install postgresql-odbc
```

- › Configure odbc.ini

```
[zabbixdb]
Description=My zabbix database
Driver=PostgreSQL Unicode
Servername = localhost
Port = 5432
Database=zabbix
UserName=zabbix
Password=HesloDoDatabaze
```



2

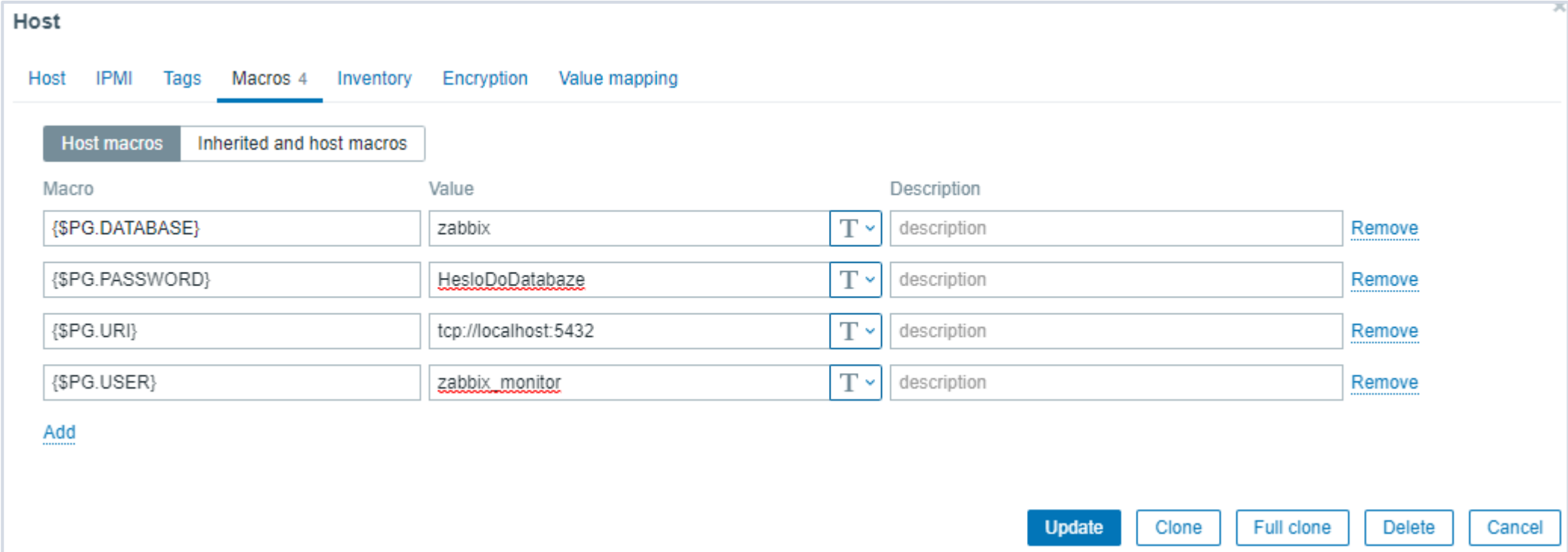
PostgreSQL Templates



Monitoring PostgreSQL with Zabbix

Official Zabbix templates

- ▶ https://git.zabbix.com/projects/ZBX/repos/zabbix/browse/templates/db/postgresql_agent2?at=release/6.4
- ▶ PostgreSQL by Zabbix agent 2



The screenshot shows the Zabbix web interface for configuring a host. The 'Host' tab is selected, and the 'Macros' sub-tab is active. The 'Host macros' section is expanded to show 'Inherited and host macros'. A table lists four macros with their values and descriptions. Each macro has a 'Remove' link to its right. At the bottom right, there are buttons for 'Update', 'Clone', 'Full clone', 'Delete', and 'Cancel'.

Macro	Value		Description	
{PG.DATABASE}	zabbix	T	description	Remove
{PG.PASSWORD}	HesloDoDatabaze	T	description	Remove
{PG.URI}	tcp://localhost:5432	T	description	Remove
{PG.USER}	zabbix_monitor	T	description	Remove

[Add](#)

[Update](#) [Clone](#) [Full clone](#) [Delete](#) [Cancel](#)

Monitoring PostgreSQL with Zabbix

Custom Queries

- ▶ Create a sql file with the query
- ▶ Set custom queries path
- ▶ `/etc/zabbix/zabbix_agent2.d/plugins.d/postgresql.conf`

```
Plugins.PostgreSQL.CustomQueriesPath=/var/lib/zabbix/
```

- ▶ Use filename as key parameter for `pgsql.custom.query`

```
pgsql.custom.query["${PG.URI}", "${PG.USER}", "${PG.PASSWORD}", "${PG.DATABASE}", "table_sizes"]
```

Monitoring PostgreSQL with Zabbix

Custom Queries

- ▶ In the template, provide the name for the sql file as the 5th parameter for the new key - `pgsql.query.custom` and specify the additional parameters for this query if needed.

* Name

Type

* Key

- ▶ `pgsql.custom.query[\"{$PG.URI}\",\"{$PG.USER}\",\"{$PG.PASSWORD}\",\"{$PG.DATABASE}\",\"table_sizes\"]`

```
/var/lib/zabbix/table_sizes.sql
```

```
Select table_schema, table_name, pg_relation_size('"' || table_schema || '"."' || table_name || "')  
from information_schema.tables order by 3
```

Advanced diagnostic using Custom Queries

- ▶ PostgreSQL diagnostic
 - ▶ pg_stat_statements
 - ▶ pg_stat_all_tables
 - ▶ ...
 - ▶ <https://www.postgresql.org/docs/current/monitoring-stats.html>

3

Demonstration – agent 2 +
PostgreSQL template



4

Important PostgreSQL metrics



Archive

- › **Count of archive files** - Total number of archived files
- › **Count of attempts to archive files** - Number of failed attempts for archiving WAL files



BGWriter

- › **Buffers allocated** - Number of buffers allocated (A buffer is an 8KB block in database shared memory.)
- › **Buffers written directly by a backend** - You want this to be small as possible
- › **Number of bgwriter stopped** - a cleaning scan because it had written too many buffers
- › **Times a backend execute its own fsync** - Number of times a backend had to execute its own fsync call

Checkpoint

- › **Buffers background written** - Should be greater than buffers written directly by a backend
- › **Buffers checkpoints written** - Should be greater than buffers written directly by a backend
- › **By timeout** - These are ok
- › **Requested** - You should be warned about these
- › **Checkpoint write time** - How much time checkpoint spend writing to disk
- › **Checkpoint sync time** - Average sync time should be near zero

Monitoring PostgreSQL with Zabbix

DBStat

- › **Blocks read time** - Time spent reading data file blocks by backends
- › **Blocks write time** - Time spent writing data file blocks by backends
- › **Checksum failures** - You should be notified about this
- › **Committed transactions** - Number of transactions that have been committed
- › **Conflicts** - Conflicts occur only on standby servers
- › **Deadlocks** - Number of deadlocks detected
- › **Disk blocks read** - From disk
- › **Hit blocks read** - From shared buffers
- › **Number temp bytes** - Total amount of data written to temporary files by queries
- › **Number of temp files** - Number of temporary files created by queries
- › **Roll backed transactions** - Number of transactions that have been rolled back

Monitoring PostgreSQL with Zabbix

DBStat

- › **Rows deleted** - Number of rows deleted by queries
- › **Rows returned** - Number of rows returned by queries
- › **Rows fetched**
 - › `select count(*) from ten_row_table`, ten row will be returned, but only one row will be fetched
- › **Rows inserted** - Number of rows inserted by queries
- › **Rows updated** - Number of rows updated by queries
- › **Backends connected** - Number of connected backends

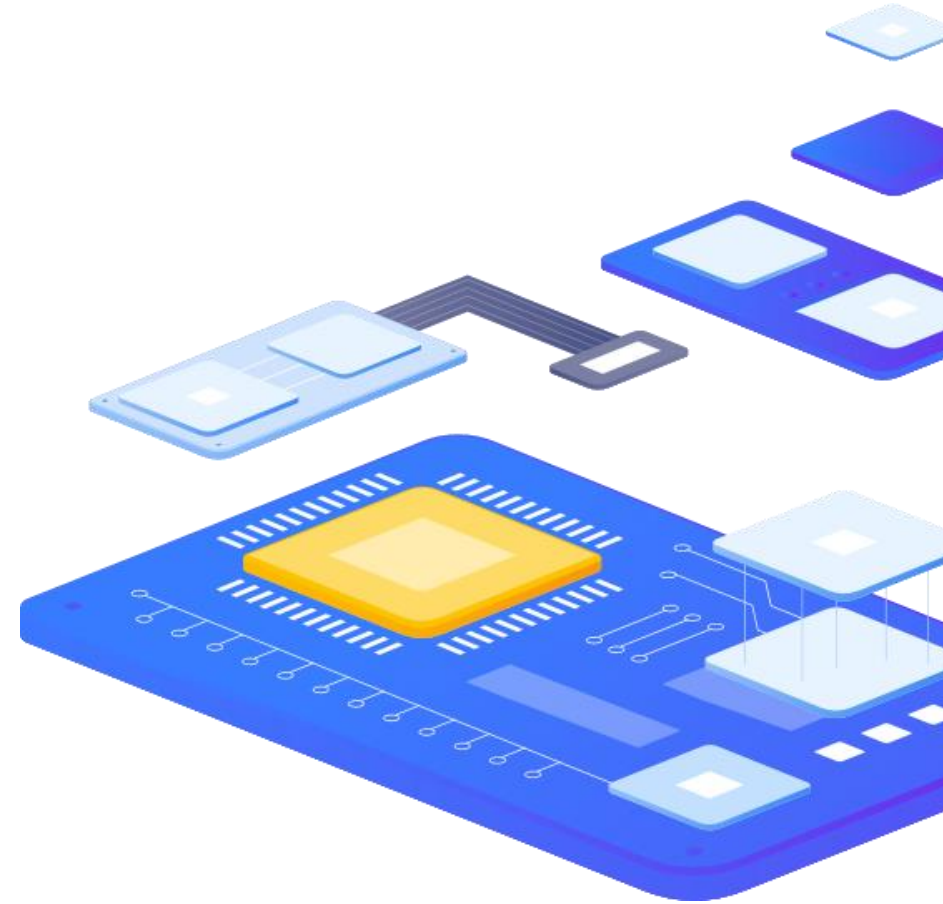
Connections

- › **Active** - Total number of connections executing a query
- › **Idle** - Total number of connections waiting for a new client command
- › **Idle in transaction** - These are problematic
- › **Prepared** - Total number of prepared transactions
- › **Total** - Total number of connections
- › **Total %** - Total number of connections in percentage
- › **Waiting** - Total number of connections waiting for connection
- › **Disabled** - Total number of disabled connections

Monitoring PostgreSQL with Zabbix

WAL

- › **Bytes written** - WAL write in bytes
- › **Segments count** - Number of WAL segments

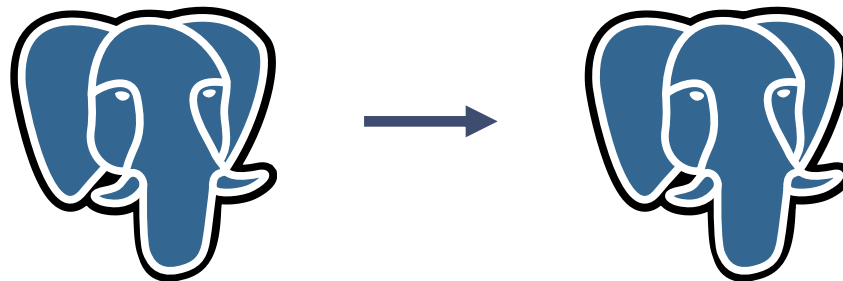


Monitoring PostgreSQL with Zabbix

Replication

- › **Lag in bytes** - Replication lag with Master in bytes.
- › **Lag in seconds** - Replication lag with Master in seconds.

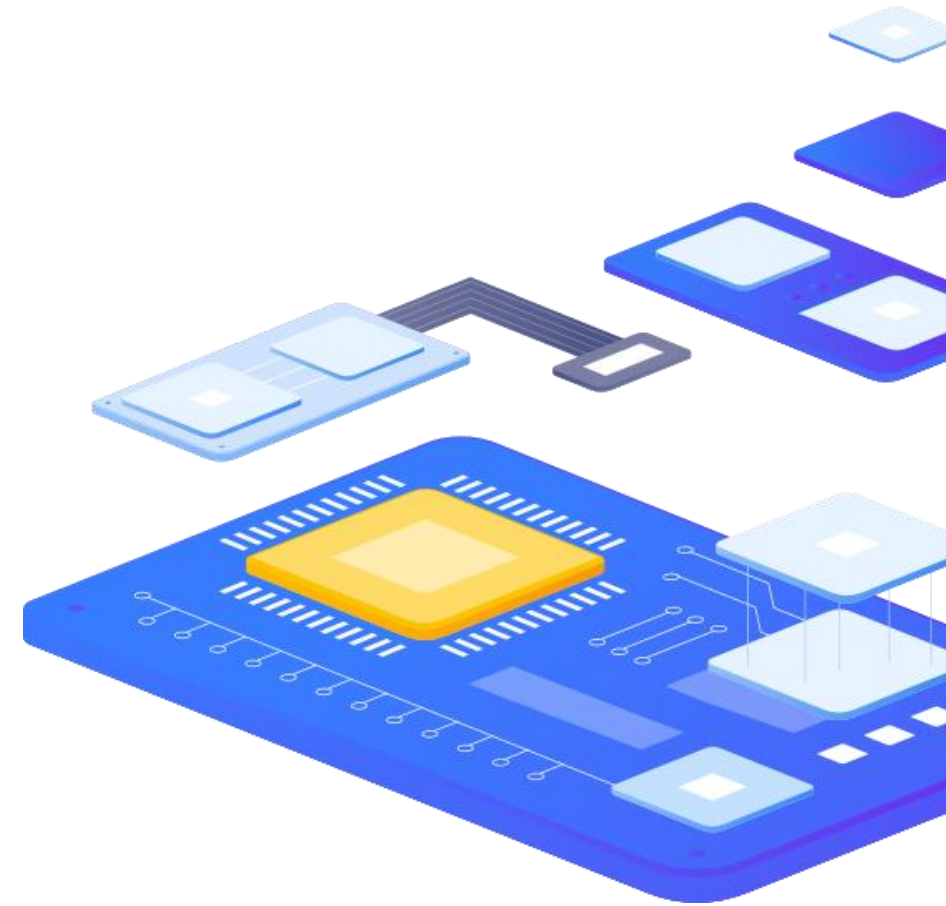
- › **Recovery role** - 0 master, 1 standby.
- › **Standby count** - Number of standby servers.
- › **Status** - 0 streaming is down, 1 streaming is up, 2 master mode



Monitoring PostgreSQL with Zabbix

OS

- › Memory - performance, stability
- › CPU - performance, stability
- › HDD - performance, stability
- › NET - performance, stability



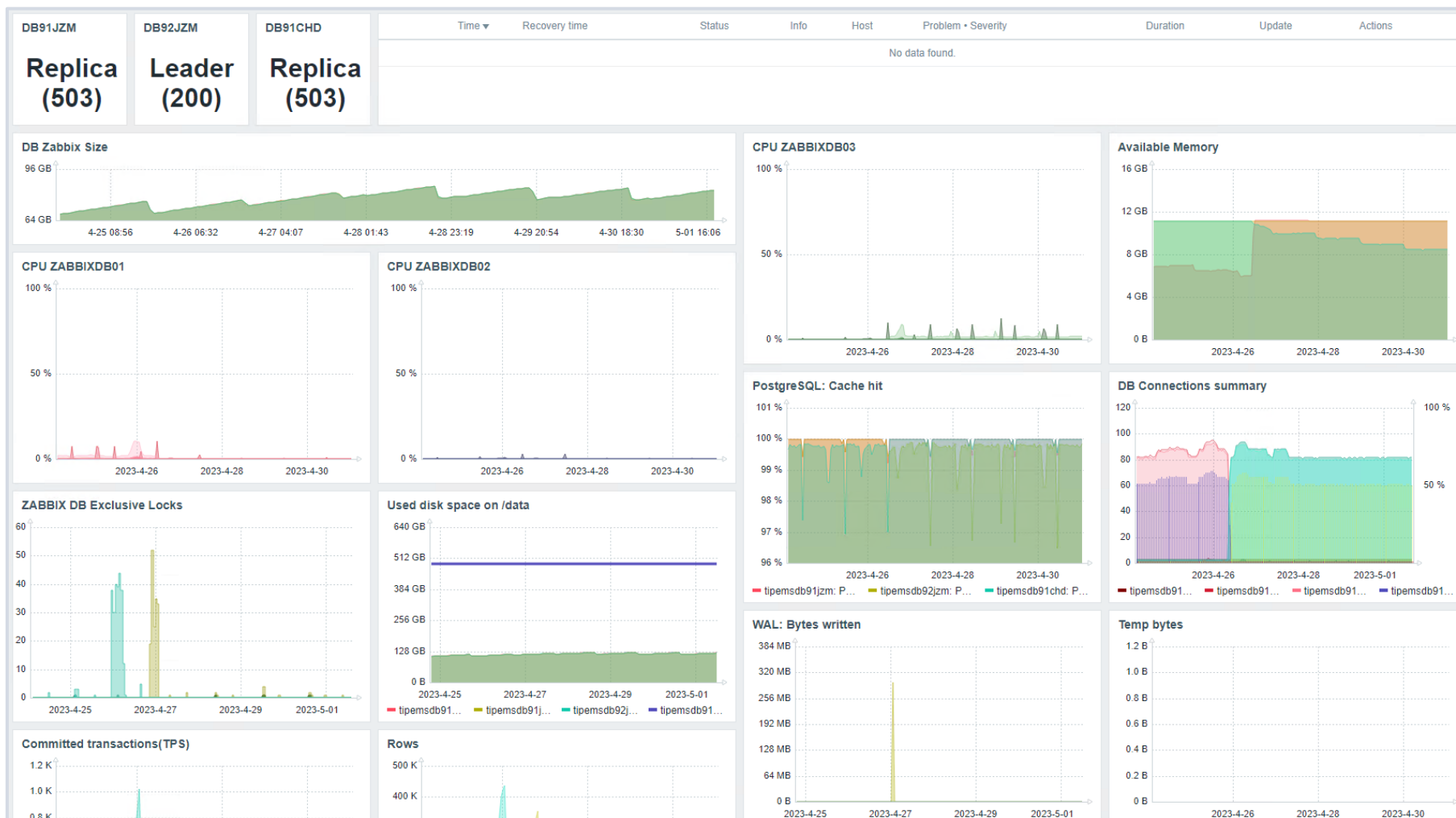
5

PostgreSQL data visualization



Monitoring PostgreSQL with Zabbix

PostgreSQL Dashboard




Monitoring PostgreSQL with Zabbix

PGWatch2

- › Who is watching the watcher?
- › Superb overview of all database activities on a single page.
- › PGWatch provides high flexibility.
- › Intuitive metrics presentation.
- › Easy to deploy.



Instance state PRIMARY	Instance uptime 2 weeks	PG version 13.6 (Ubuntu 13.6-1.pgdg20.04+1)	Shared Buffers hit pct. 99.5%
Active sessions 106	Max. connections 801	Blocked sessions 0	'Idle in TX' count 0
TX error pct. (avg.) 0.3%	TPS (avg.) 1.6 K	QPS (avg.) N/A	Query runtime (avg.) N/A
DB size 774.9 GiB	DB size change 659.1 MiB	DATADIR disk space left 0	Longest query runtime 1 second
Config change events 0	Table changes 0	WAL archiving status OK	WAL per second (avg.) 2.0 MiB
Invalid indexes 0	Duplicate indexes N/A	Checkpoints requested 0	Approx. bloat 55.6 GiB
INSERT-s per minute (avg.) 226 K	UPDATE-s per minute (avg.) 48 K	DELETE-s per minute (avg.) 15	Seq. scans on >100 MB tables per minute (avg.) 5.2
CPU load (avg.) 1.0	Temp. bytes per second (avg.) 0 B	Longest AUTOVACUUM duration N/A	Backup duration N/A
Max. table FREEZE age 199.9 Mil	Max. XMIN horizon Only queries that return single series/table	Inactive repl. slots 0	Max. replication lag N/A

Brought to you by:  **CYBERTEC**
DATA SCIENCE & POSTGRESOL



Questions?



CONTACT US:

Phone:

[+420 800 244 442](tel:+420800244442)

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](tel:+420732447184)