

PostgreSQL Monitoring Day

with Zabbix &
Postgres Professional

ONLINE



Daria Vilkova

Software Engineer
(Postgres Professional)



ZABBIX

PostgresPro

Setting Up Zabbix
Agent 2 for PostgreSQL
Monitoring and Revealing
How It Works

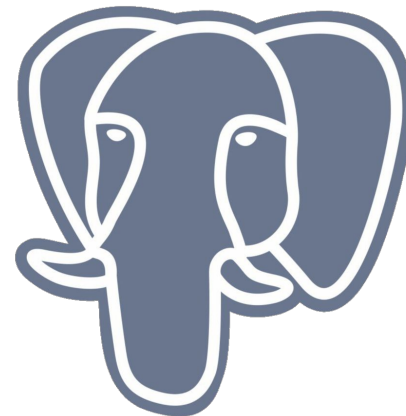
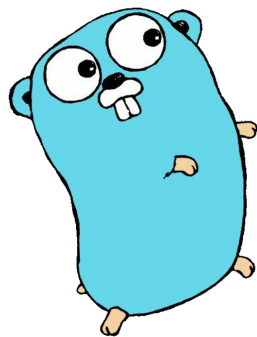
6:20 PM CET

Who am I?

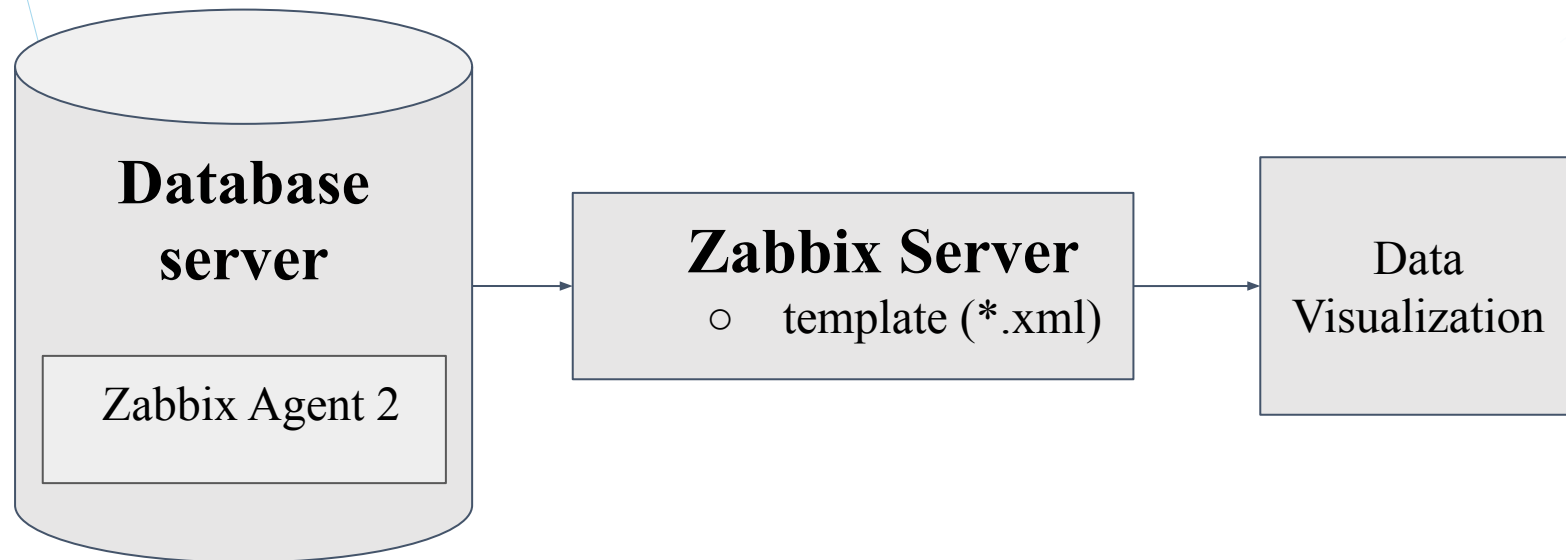
- Software Engineer at Postgres Professional for 2+ years
- Co-author of PostgreSQL monitoring plugin for Zabbix Agent 2
- Co-maintainer of mamonsu, an open-source active monitoring agent for PostgreSQL and OS based on Zabbix
- MSc Degree in Applied Mathematics and Computer Science from Lomonosov Moscow State University

What will we discuss today?

- Plugin implementation and basic capabilities
- Connection levels
- How to get a simple metric?
- How to set up custom metrics?
- Q&A



How does Zabbix work?



Implementation

- github.com/jackc/pgx - PG driver and toolkit for Go
- A handler for each metric or a group of metrics
- Some metrics are generated in JSON and grouped as dependency items and discovery rules.

Zabbix Agent 2: Basic Features

- Keeps permanent connection with **PostgreSQL**.
- Provides flexible polling intervals.
- Is compatible with **PostgreSQL** version **10+**, and **Zabbix Server** version **4.4+**.
- Is able to monitor several **PostgreSQL** instances by one Agent,

Levels of Connection Parameters

- Global
- Macros
- Sessions

Connection Parameters - Macros Level

- Fill in the template

{\$PG.URI}	tcp://localhost:5433	T ▾
{\$PG.USER}	my_user	T ▾
{\$PG.DATABASE}	my_database	T ▾

Add

Update

Clone

Full clone

Delete

Delete and clear

Cancel

Connection Parameters - Macros Level

* Name

Type

* Key

Type of information

* Update interval


Custom intervals

Type	Interval	Period	Action
<input checked="" type="checkbox"/> Flexible	<input type="text" value="Scheduling"/>	<input type="text" value="50s"/>	<input type="text" value="1-7,00:00-24:00"/>

[Add](#)

* History storage period Do not keep history Storage period

New application



Connection Parameters - Sessions Level

```
# Mandatory: no
# Default:
# Plugins.Postgres.Sessions.Test.Uri=tcp://localhost:5432

### Option: Plugins.Postgres.Sessions.*.User
# Username for session connection. "*" should be replaced with a session name.
#
# Mandatory: no
# Range: Must matches PostgreSQL user name.
# Default:
# Plugins.Postgres.Sessions.Test.User=TestUser

### Option: Plugins.Postgres.Sessions.*.Password
# Password for session connection. "*" should be replaced with a session name.
#
# Mandatory: no
# Range: Must matches the Password format.
# Default:
# Plugins.Postgres.Sessions.Test.Password=TestPassword

### Option: Plugins.Postgres.Sessions.*.Database
# Database for session connection. "*" should be replaced with a session name.
#
# Mandatory: no
# Default:
# Plugins.Postgres.Sessions.Test.Database=TestDatabase
```

Filled in
zabbix_agent2.conf

Connection Parameters - Sessions Level

Macros 7

Template macros

Inherited and template macros

Macro	Value	
{PG.CONFLICTS.MAX.WARN}	0	T ▾
{PG.CONN_TOTAL_PCT.MAX.WARN}	90	T ▾
{PG.DEADLOCKS.MAX.WARN}	0	T ▾
{PG.LLD.FILTER.DBNAME}	(.+)	T ▾
{PG.SESSION}	Test	T ▾



Connection Parameters - Sessions Level

Preprocessing

* Name

Type

* Key

Type of information

Units

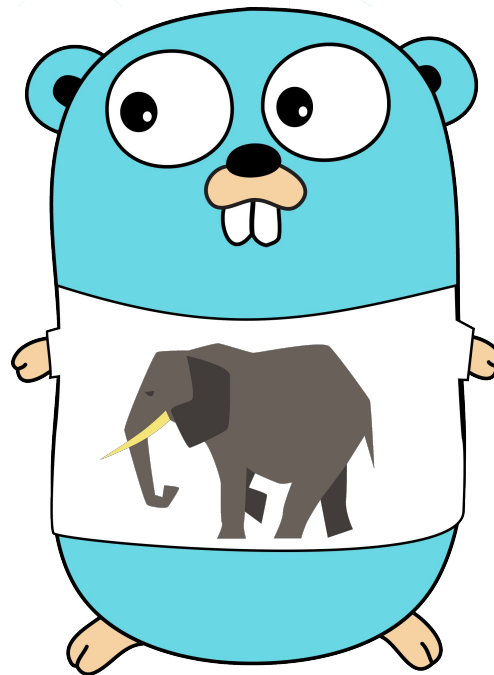
* Update interval

Custom intervals

Type	Interval	Period	Action
<input checked="" type="checkbox"/> Flexible <input type="checkbox"/> Scheduling	<input type="text" value="50s"/>	<input type="text" value="1-7,00:00-24:00"/>	Remove
Add			

Monitoring plugin in numbers

> 98 metrics



Metrics

- number of connections
- database size
- info about archive files
- number of “bloating” tables
- replication status
- background writer processes activity
- ...

Metrics

<input type="checkbox"/> Host	Name ▲	Interval	History
▶ go-agent-demo	CPU (15 Items)		
▶ go-agent-demo	General (5 Items)		
▶ go-agent-demo	Memory (5 Items)		
▶ go-agent-demo	OS (8 Items)		
▶ go-agent-demo	Performance (15 Items)		
▶ go-agent-demo	PostgreSQL (55 Items)		
▶ go-agent-demo	PostgreSQL: DB pgbench_1gb (3 Items)		
▶ go-agent-demo	PostgreSQL: DB pgbench_3gb (3 Items)		
▶ go-agent-demo	PostgreSQL: DB pgbench_10gb (3 Items)		

Example: How to Get a Simple Metric?

Create a file to get a new metric:

zabbix/src/go/plugins/postgres/handler_uptime.go

Import package and specify key for new metric:

```
package postgres
```

```
const (  
    keyPostgresUptime = "pgsql.uptime"  
)
```


Example: How to Get a Simple Metric?

Create a handler with the following query:

```
func uptimeHandler(ctx context.Context, conn PostgresClient, _ string, _  
map[string]string, _ ...string) (interface{}, error) {
```

```
    var uptime float64
```

```
    query := `SELECT date_part('epoch', now()) - pg_postmaster_start_time();`
```

Example: How to Get a Simple Metric?

Run the following query:

```
row, err := conn.QueryRow(ctx, query)
```

```
if err != nil {
```

```
    ...
```

```
}
```

```
err = row.Scan(&uptime)
```

```
if err != nil {
```

```
    ...
```

```
}
```

```
return uptime, nil
```

Example: How to Get a Simple Metric?

Register the key of your new metric in metrics.go:

```
var metrics = metric.MetricSet{  
    .....,  
    keyPostgresUptime: metric.New("Returns uptime.",  
    []*metric.Param{paramURI, paramUsername,  
paramPassword,paramDatabase}, false),  
}
```

Example: How to Get a Simple Metric?

Recompile the agent!



Coming soon: custom metrics feature

Custom Metrics

- Create a sql file with the query.
- In `zabbix_agent2.conf` specify the path to the directory with the sql files named **Plugins.Postgres.CustomQueriesPath**.
- 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.

Custom Metrics

- # CREATE table example (phrase text, year int);
- # SELECT * FROM example;

```
phrase          | year
-----+-----
```

```
new 2020 year is coming | 2020
```

```
new 2021 year is coming | 2021
```

```
new 2022 year is coming | 2022
```

(3 rows)

Custom Metrics

- `$touch custom2.sql`
- `$echo "SELECT * FROM example;" > custom2.sql`

- `$touch custom1.sql`
- `$echo "SELECT phrase FROM example WHERE year=$1;" > custom1.sql`

Custom Metrics

- Add path to the sql files in **zabbix_agent2.conf**

Plugins.Postgres.CustomQueriesPath=/path/to/file

Custom Metrics

- `pgsql.query.custom - key`

* Name

Type

* Key

Type of information

* Update interval

Custom intervals

Type	Interval	Period	Action
<input type="text" value="Flexible"/> <input type="text" value="Scheduling"/>	<input type="text" value="50s"/>	<input type="text" value="1-7,00:00-24:00"/>	Remove
Add			

* History storage period

Custom Metrics

- `pgsql.query.custom` - key

* Name

Type

* Key

Type of information

* Update interval

Custom intervals

Type	Interval	Period	Action
<input type="text" value="Flexible"/> <input type="text" value="Scheduling"/>	<input type="text" value="50s"/>	<input type="text" value="1-7,00:00-24:00"/>	Remove
Add			

* History storage period

Custom Metrics

The result for each query will appear in text format.

Custom queries 	30s	90d	Zabbix agent	2020-11-29 20:53:53	[{"phrase":"new 2020 year is ...
pgsql.custom.query["unix:/tmp/.s.PGSQL.543...					
Custom queries with parameters 	30s	90d	Zabbix agent	2020-11-29 20:53:52	[{"phrase":"new 2021 year is ...
pgsql.custom.query["unix:/tmp/.s.PGSQL.543...					

Plugin Availability

The new version with custom metrics
will become available with the next

Zabbix Server release:

<https://www.zabbix.com/download>

Useful Links

- Zabbix Git :

<https://git.zabbix.com/projects/ZBX/repos/zabbix/browse/src/go/plugins/postgres>

- Official template of the plugin:

<https://www.zabbix.com/ru/integrations/postgresql#tab:official2>

- PostgreSQL monitoring plugin article:

<https://postgrespro.com/blog/pgsql/5967895>

**Thank you for
your attention!**

info@postgrespro.com

postgrespro.com

d.vilkova@postgrespro.com

Q&A

