

Мониторинг PostgreSQL с использованием Zabbix

Вилкова Дарья

Postgres Professional

План доклада

- Mamonsu - агент мониторинга PostgreSQL и ОС
- Модуль мониторинга PostgreSQL в составе Zabbix Agent 2

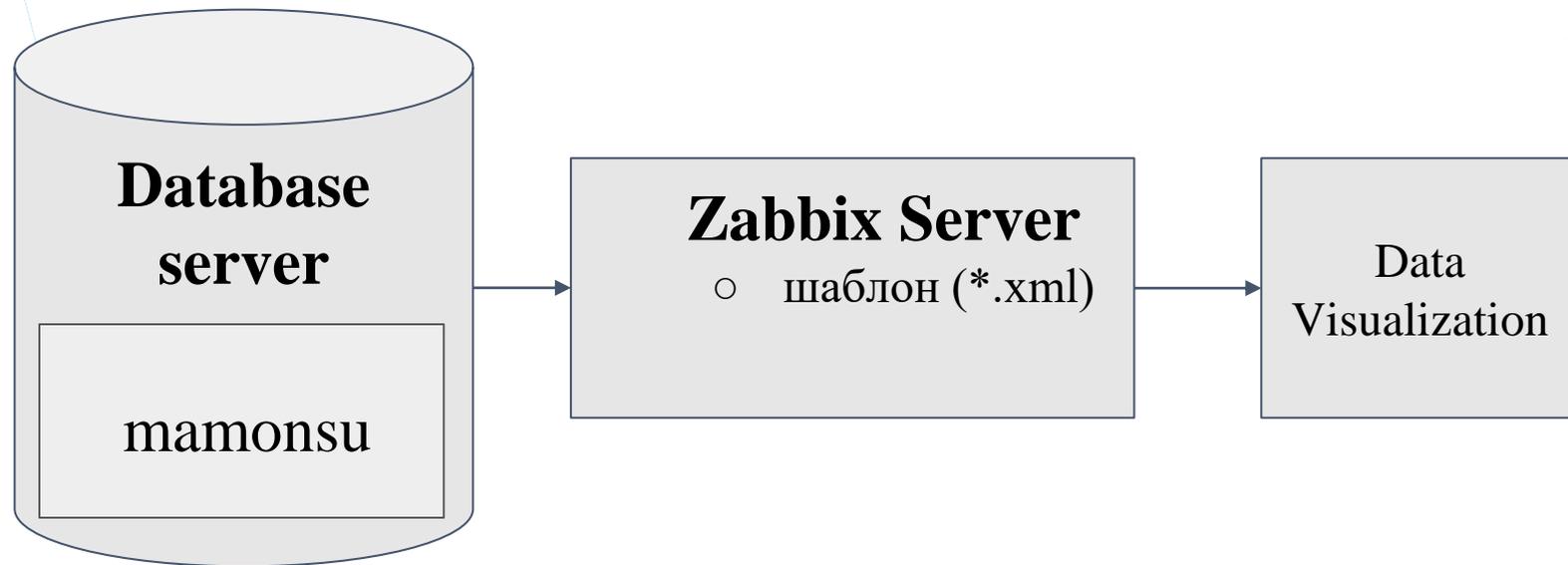
Mamonsu



Что такое Mamonsu

- Активный агент (Zabbix Trapper) мониторинга для PostgreSQL и операционной системы
- Обладает дополнительными инструментами:
 - mamonsu tune
 - mamonsu report
- Написан на Python

Схема работы



Возможности Mamonsu

- Эффективная работа с PostgreSQL
- Расширяемость:
 - можно самому писать плагины
 - возможность настроить параметры сбора метрик под себя
- Широкий охват метрик для мониторинга, специфичных для PostgreSQL
- Работа “из коробки”
- Выгружает шаблоны и конфигурационные файлы
- Кроссплатформенный
- BSD 3-clause licence

Матонсу в цифрах

- 14 плагинов для PostgreSQL
- 8 плагинов для OS Linux
- 4 плагинов для OS Windows

Матрица в цифрах

> 70 метрик PostgreSQL

> 40 метрик OS Linux

> 8 метрик OS Windows

В сумме более 110 метрик

Основные метрики

- Доступность СУБД
- Количество соединений
- Размеры баз данных
- Чекпоинты
- Скорость чтения / записи
- Блокировки
- Количество процессов автовакуум
- Скорость генерации WAL

Metrics: PostgreSQL

```
'PostgreSQL: ping': pgsql.ping[]
'PostgreSQL: service uptime': pgsql.uptime[]
'PostgreSQL: cache hit ratio': pgsql.cache[hit]
'PostgreSQL: number of total connections': pgsql.connections[total]
'PostgreSQL: number of waiting connections': pgsql.connections[waiting]
'PostgreSQL: number of active connections': pgsql.connections[active]
'PostgreSQL: number of idle connections': pgsql.connections[idle]
'PostgreSQL: number of idle in transactions connections': pgsql.connections[idle_in_transaction]
'PostgreSQL: number of idle in transactions aborted connections': pgsql.connections[idle_in_transaction_aborted]
'PostgreSQL: number of fastpath function call connections': pgsql.connections[fastpath_function_call]
'PostgreSQL: number of disabled connections': pgsql.connections[disabled]
'PostgreSQL: number of max connections': pgsql.connections[max_connections]
'PostgreSQL: count files in archive_status need to archive': pgsql.archive_command[count_files_to_archive]
'PostgreSQL: size of files need to archive': pgsql.archive_command[size_files_to_archive]
'PostgreSQL: count archived files': pgsql.archive_command[archived_files]
'PostgreSQL: count attempts to archive files': pgsql.archive_command[failed_trying_to_archive]
'PostgreSQL checkpoint: by timeout (in hour)': pgsql.checkpoint[count_timed]
'PostgreSQL checkpoint: by wal (in hour)': pgsql.checkpoint[count_wal]
'PostgreSQL checkpoint: write time': pgsql.checkpoint[write_time]
'PostgreSQL checkpoint: sync time': pgsql.checkpoint[checkpoint_sync_time]
'PostgreSQL bgwriter: buffers written during checkpoints': pgsql.bgwriter[buffers_checkpoint]
'PostgreSQL bgwriter: buffers written': pgsql.bgwriter[buffers_clean]
'PostgreSQL bgwriter: number of bgwriter stopped by max write count': pgsql.bgwriter[maxwritten_clean]
'PostgreSQL bgwriter: buffers written directly by a backend': pgsql.bgwriter[buffers_backend]
```

<https://github.com/postgrespro/mamonsu>



Запуск Mamonsu за 5 минут

- Установка Mamonsu:

```
$ git clone ... && cd mamonsu && python setup.py  
build && python setup.py install
```

- Настройка соединений:

```
/etc/mamonsu/agent.conf
```

- Экспорт шаблона в Zabbix Server:

```
$ mamonsu zabbix template export  
/usr/share/mamonsu/example.xml
```

- Добавление хоста в Zabbix Server:

```
$ mamonsu zabbix host create mamonsu-demo
```

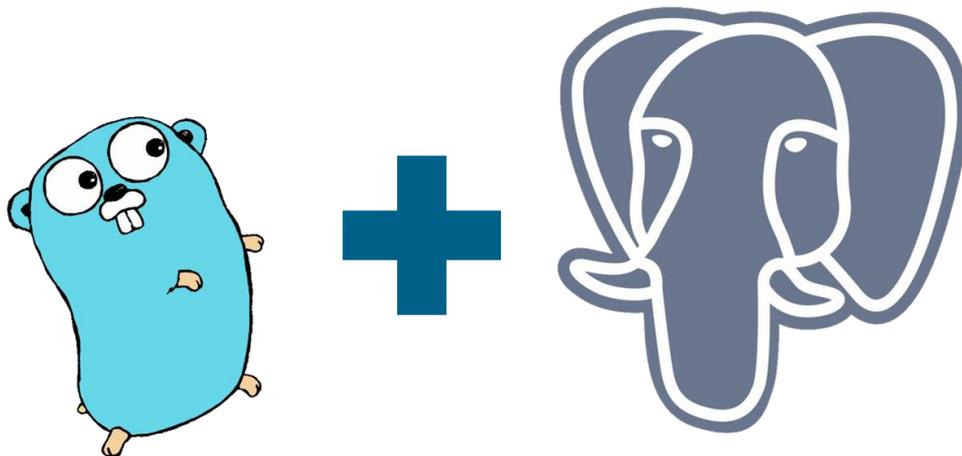
- Запуск:

```
$ service mamonsu start
```

Направления развития Mamonsu

- Новые метрики и плагины
- Плагин мониторинга размера отдельных таблиц
- Расширение возможностей авто-настройки через *mamonsu tune*

Модуль мониторинга PostgreSQL в составе Zabbix Agent 2



Реализация

- github.com/jackc/pgx - PG driver and toolkit for Go
- подключение интерфейсов Exporter, Configurator
Zabbix Agent 2
- обработчик (handler) для метрики или группы метрик
- получение части метрик по группам в JSON как
dependency items и discovery rules

Основные возможности

- сохранение постоянного соединения с **PostgreSQL** между проверками
- поддержка гибких интервалов опроса
- совместимость с версиями **PostgreSQL**, начиная с 10, и **Zabbix Server**, начиная с 4.4
- возможность подключения к нескольким экземплярам **PostgreSQL** одновременно

Уровни параметров подключения к PostgreSQL

- Global
- Sessions
- Macros

Параметры подключения к PostgreSQL - Sessions

- Заполняются в `zabbix_agent2.conf`

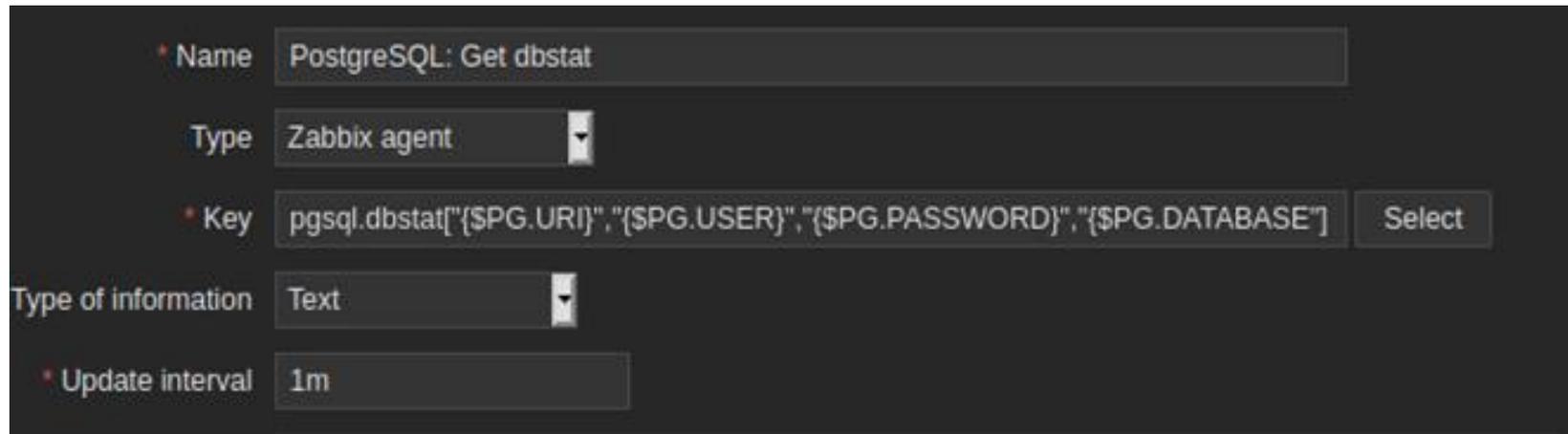
```
### Option: Plugins.Postgres.Sessions.*.Uri
#   Connection string. "*" should be replaced with a session name.
#
# Mandatory: no
# Default:
Plugins.Postgres.Sessions.Test.Uri=tcp://localhost:5432

### 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=TestUserName

### 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
```

Параметры подключения к PostgreSQL - Marcos

- Указываются в ключе метрики в шаблоне



The image shows a configuration form for a Zabbix agent item. The fields are as follows:

- Name:** PostgreSQL: Get dbstat
- Type:** Zabbix agent
- Key:** pgsql.dbstat["{\$PG.URI}", "{\$PG.USER}", "{\$PG.PASSWORD}", "{\$PG.DATABASE}"]
- Type of information:** Text
- Update interval:** 1m

A "Select" button is located to the right of the Key field.

Модуль мониторинга PostgreSQL в цифрах

> 95 метрик

Метрики

- количество соединений
- объем баз данных
- архивация wal файлов
- контрольные точки
- количество “раздувшихся” таблиц
- статус репликации
- отставание реплики
- ...

Метрики

<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)		

Пример получения простой метрики

Создаем файл для получения новой метрики:

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

Подключаем пакет и указываем ключ (ключи) метрик:

```
package postgres
```

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

Пример получения простой метрики

Создаем обработчик (handler) с запросом:

```
func (p *Plugin) uptimeHandler(conn *postgresConn, params []string)(interface{},
error)
{
    var uptime float64
    query := `SELECT date_part('epoch', now() - pg_postmaster_start_time());`
```

Пример получения простой метрики

Выполняем запрос:

```
err := conn.postgresPool.QueryRow(context.Background(),
query).Scan(&uptime)
  if err != nil {
    ...
  }
return uptime, nil
```

Пример получения простой метрики

Регистрируем ключ новой метрики:

```
func init() {  
    plugin.RegisterMetrics(&impl, pluginName,  
        ...  
        keyPostgresUptime, "Returns uptime.",  
    )  
}
```

Собираем агент!

Как попробовать ?

Доступен в Zabbix 5.0.1

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

Ссылки

- GitHub mamonsu:
<https://github.com/postgrespro/mamonsu>
- Документация mamonsu:
<https://postgrespro.ru/docs/postgrespro/11/mamonsu>
- Zabbix Git :
<https://git.zabbix.com/projects/ZBX/repos/zabbix/browse/src/go/plugins/postgres>

Спасибо за внимание!



7А

Москва, ул. Дмитрия Ульянова



+7 (495) 150-06-91



info@postgrespro.ru

d.vilkova@postgrespro.ru

www.postgrespro.ru

Вопросы?

