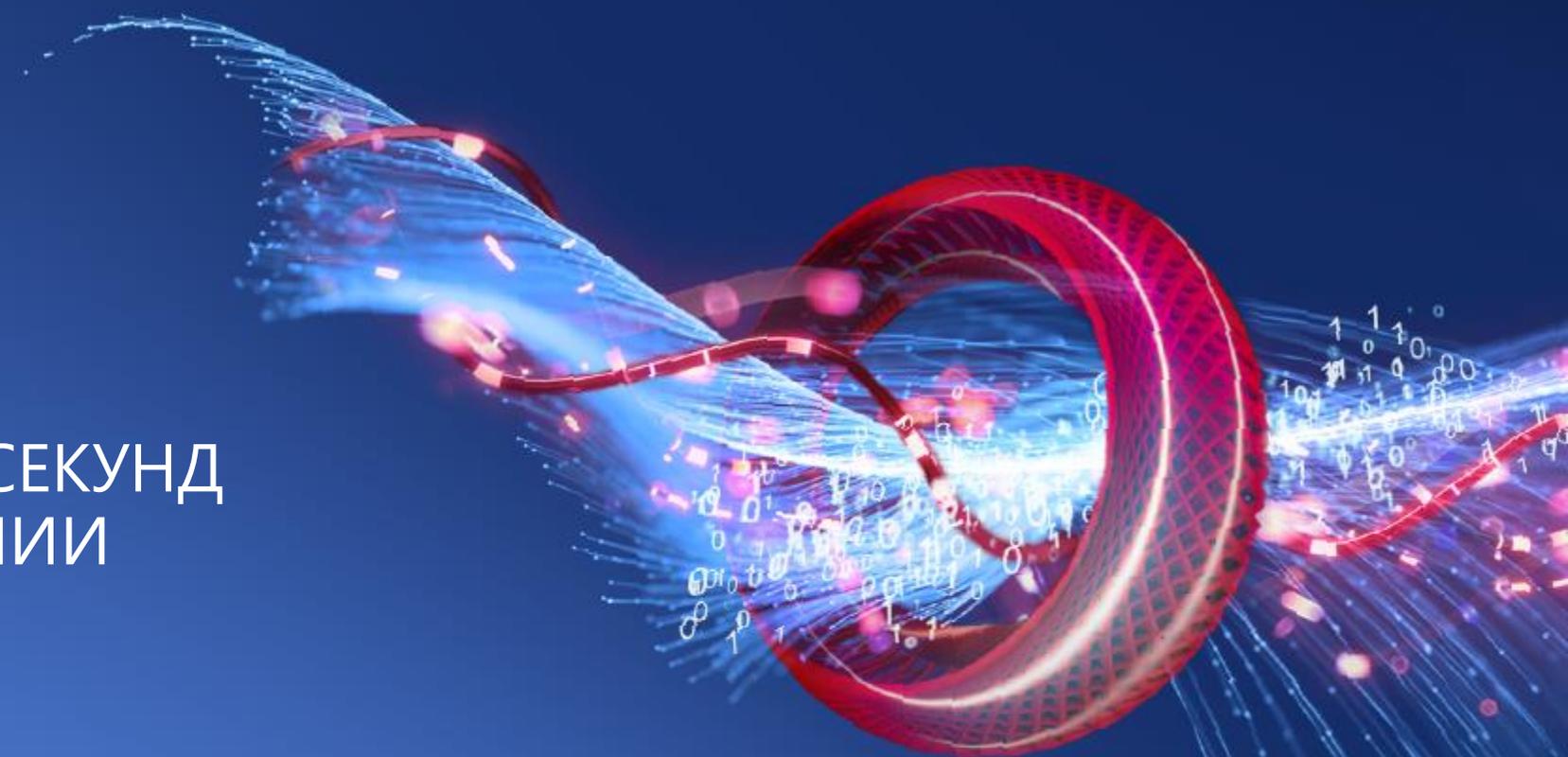




**ZABVIX** 5.0

5 ПРОСТЫХ СПОСОБОВ  
РЕШЕНИЯ  
СЛОЖНЫХ ЗАДАЧ

# 01



## ПОДДЕРЖКА НАНОСЕКУНД ПРИ ИСПОЛЬЗОВАНИИ ZABBIX\_SENDER

- ⦿ Новая опция `zabbix_sender`
- ⦿ Поддержка наносекунд во входящем файле Zabbix sender. Может быть использована вместе с опцией `-T --with-timestamps`
- ⦿ Опция подразумевает наличие в каждой строке входящего файла разделённых пробелами данных вида
- ⦿ `<host> <key> <timestamp> <ns> <value>`

# ZABBIX SENDER

## ЧТО ЭТО И КАК ИСПОЛЬЗОВАТЬ ( ДО 5.0 )

- Легковесная, простая в установке утилита
- Используется для отправки данных траппер элементам Zabbix-сервера/прокси
- Чаще всего применяется во внешних скриптах, выполняемых без участия Zabbix

```
shell> cd bin
shell> ./zabbix_sender -z zabbix -s "Linux DB3" -k db.connections -o 43
```

- Применение в высокочастотный мониторинге
- Поддержка входящих файлов--input-file ; -i
- Поддержка меток времени --with-timestamps ; -T

```
Input file format
<hostname> <key> <timestamp> <value>
```

# ZABBIX SENDER

## В ЧЁМ МОЖЕТ ВОЗНИКНУТЬ ПРОБЛЕМА?

- Некоторые решения для репликации баз данных требуют использования первичных ключей во всех таблицах ( Galera, Innodb Cluster )
- В базе данных Zabbix есть таблицы без первичных ключей ( history ...)
- Структура таблицы "history" по умолчанию:

```
MariaDB [zabbix]> show create table history;
+-----+
| Table | Create Table
+-----+
| history | CREATE TABLE `history` (
  `itemid` bigint(20) unsigned NOT NULL,
  `clock` int(11) NOT NULL DEFAULT '0',
  `value` double(16,4) NOT NULL DEFAULT '0.0000',
  `ns` int(11) NOT NULL DEFAULT '0',
  KEY `history_1` (`itemid`,`clock`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin |
+-----+
1 row in set (0.00 sec)
```

- Как же добиться этого? Создать первичные ключи вручную!

# ZABBIX SENDER ПРОБЛЕМА

> Alter table history add primary key (itemid, clock) – Хороший вариант? (спойлер - не очень)..

```
[root@meetup tmp]# cat /tmp/input.txt
ZabbixRocks item 1589529351 5
[root@meetup tmp]#
[root@meetup tmp]#
[root@meetup tmp]# zabbix_sender -z 127.0.0.1 -i /tmp/input.txt -T
Response from "127.0.0.1:10051": "processed: 5; failed: 0; total: 5; seconds spent: 0.000521"
sent: 5; skipped: 0; total: 5
```

```
7093:20200507:233629.509 [Z3005] query failed: [1062] Duplicate entry '28754-1589529351-0' for key 'PRIMARY' [insert into history (itemid,clock,ns,value) values (28754,1589529351,0,5.000000);
]
7093:20200507:233629.510 [Z3005] query failed: [1062] Duplicate entry '28754-1589529351-1' for key 'PRIMARY' [insert into history (itemid,clock,ns,value) values (28754,1589529351,1,5.000000);
]
7093:20200507:233629.510 [Z3005] query failed: [1062] Duplicate entry '28754-1589529351-2' for key 'PRIMARY' [insert into history (itemid,clock,ns,value) values (28754,1589529351,2,5.000000);
]
7093:20200507:233629.510 [Z3005] query failed: [1062] Duplicate entry '28754-1589529351-3' for key 'PRIMARY' [insert into history (itemid,clock,ns,value) values (28754,1589529351,3,5.000000);
]
7093:20200507:233629.510 [Z3005] query failed: [1062] Duplicate entry '28754-1589529351-4' for key 'PRIMARY' [insert into history (itemid,clock,ns,value) values (28754,1589529351,4,5.000000);
]
```

# ZABBIX SENDER

## ПОДДЕРЖКА НАНОСЕКУНД

- Используйте более подходящий метод формирования первичных ключей
- Alter table history add primary key (itemid, clock,ns) – **Хороший вариант!**
- Используйте дополнительные опции Zabbix-sender -N ; --with-ns

```
Input file format
<hostname> <key> <timestamp> <nanoseconds> <value>
```

```
[root@meetup tmp]# cat input.txt
ZabbixRocks item 1589529351 123 5
ZabbixRocks item 1589529351 234 5
ZabbixRocks item 1589529351 345 5
ZabbixRocks item 1589529351 456 5
ZabbixRocks item 1589529351 567 5
[root@meetup tmp]#
[root@meetup tmp]#
[root@meetup tmp]# zabbix_sender -z 127.0.0.1 -i /tmp/input.txt -T --with-ns -vv
zabbix_sender [8323]: DEBUG: answer [{"response":"success","info":"processed: 5; failed: 0; total: 5; seconds spent: 0.000099"}]
Response from "127.0.0.1:10051": "processed: 5; failed: 0; total: 5; seconds spent: 0.000099"
sent: 5; skipped: 0; total: 5
```

```
MariaDB [zabbix]> select * from history where itemid = 28754;
+-----+-----+-----+-----+
| itemid | clock      | value  | ns  |
+-----+-----+-----+-----+
| 28754  | 1589529351 | 5.0000 | 123 |
| 28754  | 1589529351 | 5.0000 | 234 |
| 28754  | 1589529351 | 5.0000 | 345 |
| 28754  | 1589529351 | 5.0000 | 456 |
| 28754  | 1589529351 | 5.0000 | 567 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

# 02



## ТРИГГЕРЫ С ФУНКЦИЕЙ NODATA И ДОСТУПНОСТЬ ПРОКСИ

- ⊙ Триггеры с функцией `nodata()` по умолчанию чувствительны к доступности прокси
- ⊙ Возможность отключения чувствительности к доступности прокси при использовании нового параметра функции `nodata(5m,strict)`
- ⊙ Избегайте информационных штормов и других проблем из-за одного прокси

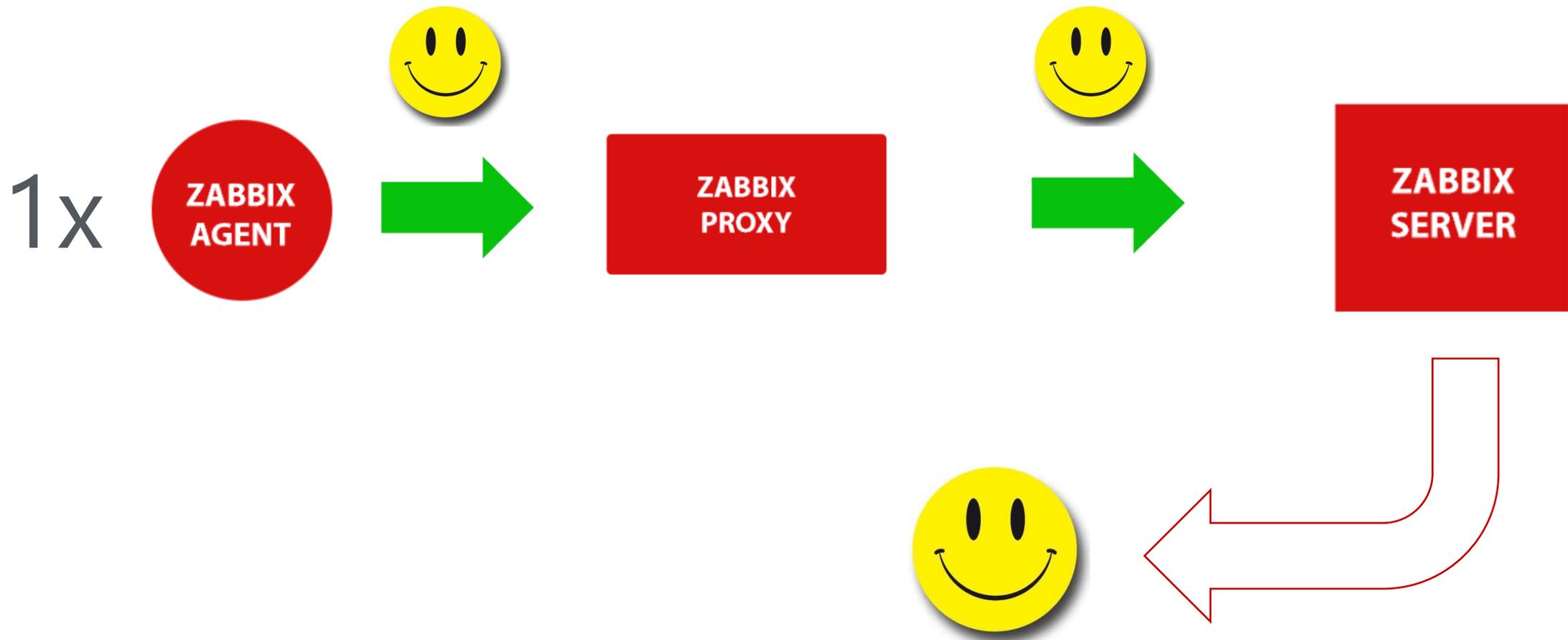
# ТРИГГЕРЫ С ФУНКЦИЕЙ NODATA

## ОПИСАНИЕ РАБОТЫ NODATA

[ZBXNEXT-1891](#) – Неявная зависимость триггера при мониторинге с прокси( 2003 )

- Функция триггеров `nodata()` – Проверка не/получения данных
- Нужно задать тип и диапазон проверки
- `host:item.nodata(5m)=1` – нет данных
- `host:item.nodata(5m)=0` – данные есть
- Проверка производится каждые 30 сек!
- Триггеры с функцией `nodata()` используются в официальных шаблонах Zabbix

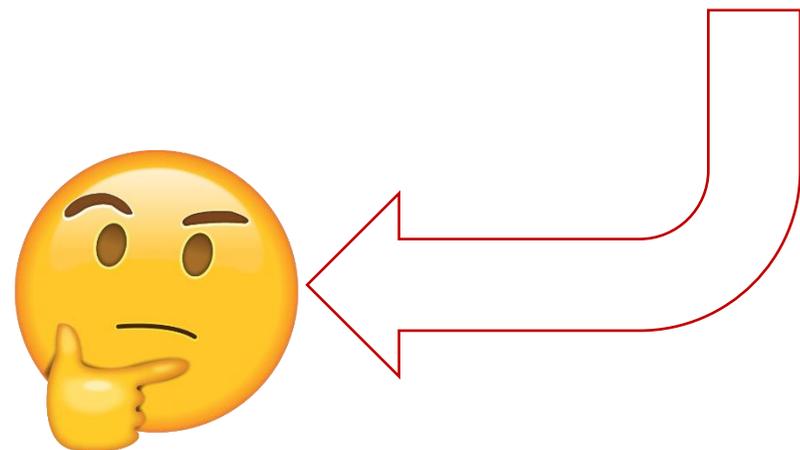
# ТРИГГЕРЫ С ФУНКЦИЕЙ NODATA МОДЕЛЬ РАБОТЫ



# ТРИГГЕРЫ С ФУНКЦИЕЙ NODATA МОДЕЛЬ РАБОТЫ



- Если хост мониторится с прокси,  
но прокси не может отправить данные серверу= no data



# ТРИГГЕРЫ С ФУНКЦИЕЙ NODATA МОДЕЛЬ РАБОТЫ

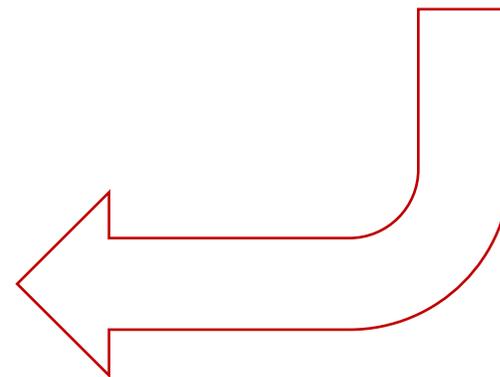


5000x проблем с nodata

Штормы событий, ложных оповещений

Возможные проблемы с производительность

Быть может – отказаться от nodata()?



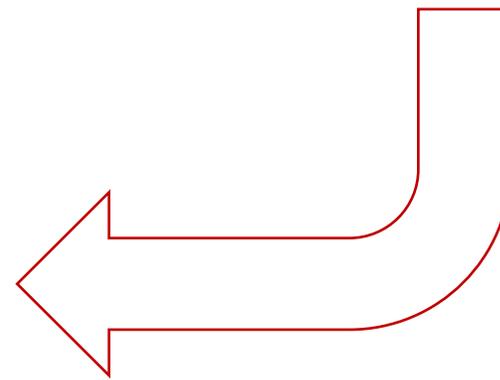
# ТРИГГЕРЫ С ФУНКЦИЕЙ NODATA МОДЕЛЬ РАБОТЫ– 5.0



Всего 1 проблема

Proxy is down

Использовать `nodata()` можно и нужно!



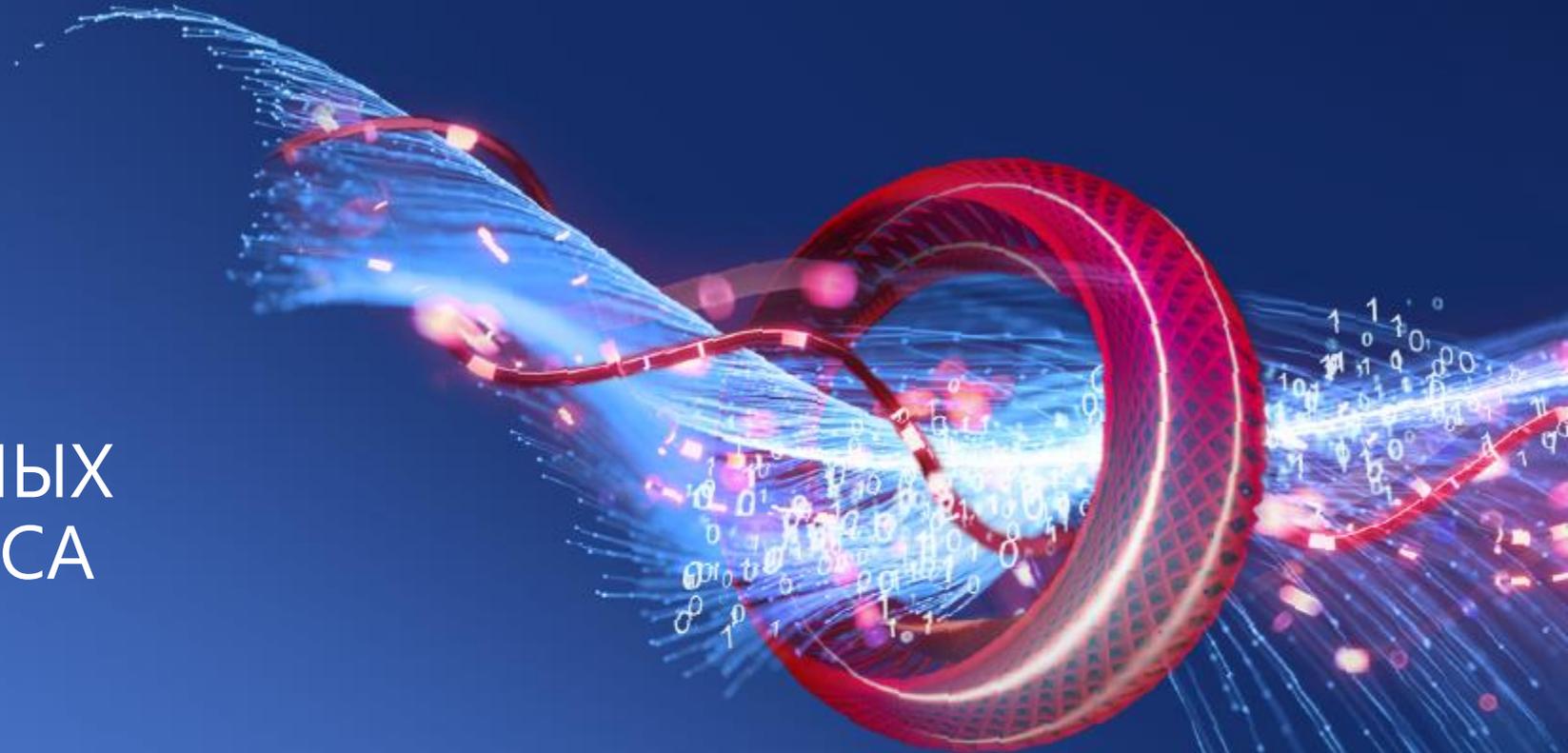
# ТРИГГЕРЫ С ФУНКЦИЕЙ NODATA В ЗАКЛЮЧЕНИИ

- Был ли способ устранения до 5.0 ?
- Глобальная корреляция
- Сколько времени нужно на настройку этой функции в 5.0 ?
- А если мне нужна зависимость от доступности прокси? Nodata(5,strict) !

# 03

## ТЕСТИРОВАНИЕ ЭЛЕМЕНТОВ ДАННЫХ ИЗ ВЕБ-ИНТЕРФЕЙСА

- ⦿ В предыдущих версиях было трудно сказать, корректен ли новый элемент данных
- ⦿ Теперь возможно протестировать элемент данных до сохранения, получив реальное значение сразу.
- ⦿ И даже протестировать все шаги препроцессинга



# ТЕСТИРОВАНИЕ ЭЛЕМЕНТОВ ДАННЫХ ЧТО МЫ ДЕЛАЛИ РАНЬШЕ?

- Просто ждали следующего интервала обновления
- `zabbix_agentd -t < item_key >`
- `Zabbix_get -s <host> -k <key>`

\* Имя

Тип

\* Ключ

\* Интерфейс узла сети

Тип информации

\* Интервал обновления

Пользовательские интервалы

Тип

[Добавить](#)

```
[root@meetup tmp]# zabbix_get -s 127.0.0.1 -k system.hostname
meetup
[root@meetup tmp]#
[root@meetup tmp]#
[root@meetup tmp]# zabbix_agentd -t system.hostname
system.hostname
[root@meetup tmp]#
```

# ТЕСТИРОВАНИЕ ЭЛЕМЕНТОВ ДАННЫХ

## ПРОБЛЕМЫ

- Интервал обновления может быть слишком большим
- В случае пользовательских интервалов – и того больше
- Zabbix\_agentd and zabbix-get работают без препроцессинга

# ТЕСТИРОВАНИЕ ЭЛЕМЕНТОВ ДАННЫХ

## 5.0

- Выберите IP-адрес и порт, с которых вы хотите проверить элемент данных
- Хотите проверить с прокси? Почему нет?
- Наблюдайте прогресс выполнения всех шагов препроцессинга
- Лёгкая настройка сложных элементов данных даже на уровне шаблона

**Test item** ✕

Get value from host

Host address  Port

Proxy

Value

Previous value

End of line sequence

Preprocessing steps

Name	Result
1: Regular expression	

# ТЕСТИРОВАНИЕ ЭЛЕМЕНТОВ ДАННЫХ

## 5.0

Test item

Get value from host

Host address  Port

Proxy

Value  Time

Previous value  Prev. time

End of line sequence  LF  CRLF

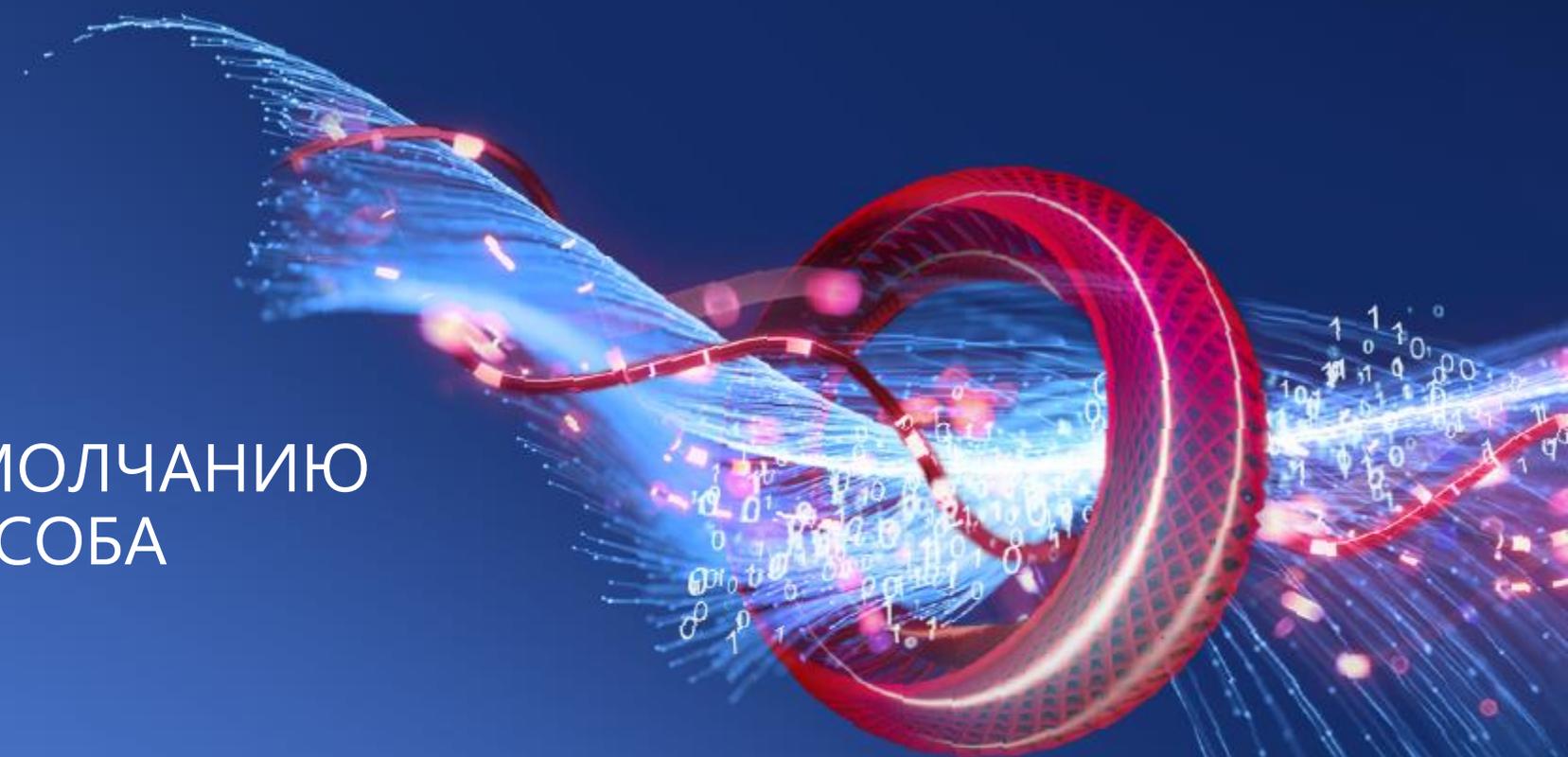
Preprocessing steps

Name	Result
1: Regular expression	demo2
2: Replace	Zabbix meetup 5.0
3: Left trim	5.0
4: Custom multiplier	10
5: In range	10

Result  10

```
[root@demo2 ~]# zabbix_get -s 127.0.0.1 -k system.hostname  
demo2.zabbix.lan  
[root@demo2 ~]#
```

# 04



## СООБЩЕНИЯ ПО УМОЛЧАНИЮ ДЛЯ КАЖДОГО СПОСОБА ОПОВЕЩЕНИЙ

- ☉ Управлять текстом в сообщениях стало проще
- ☉ Массовое применение изменений в действиях для оповещений в два клика
- ☉ Упрощение настройки действий

# СООБЩЕНИЯ ПО УМОЛЧАНИЮ ДЛЯ КАЖДОГО СПОСОБА ОПОВЕЩЕНИЙ

- Ранее настраивались для каждого действия
- Достаточно гибко, но...
- Необходимость в поддержке большого количества действий
- Трудности в массовой корректировке

### Actions

Action Operations Recovery operations Update operations

\* Default operation step duration

Default subject

Default message

Pause operations for suppressed problems

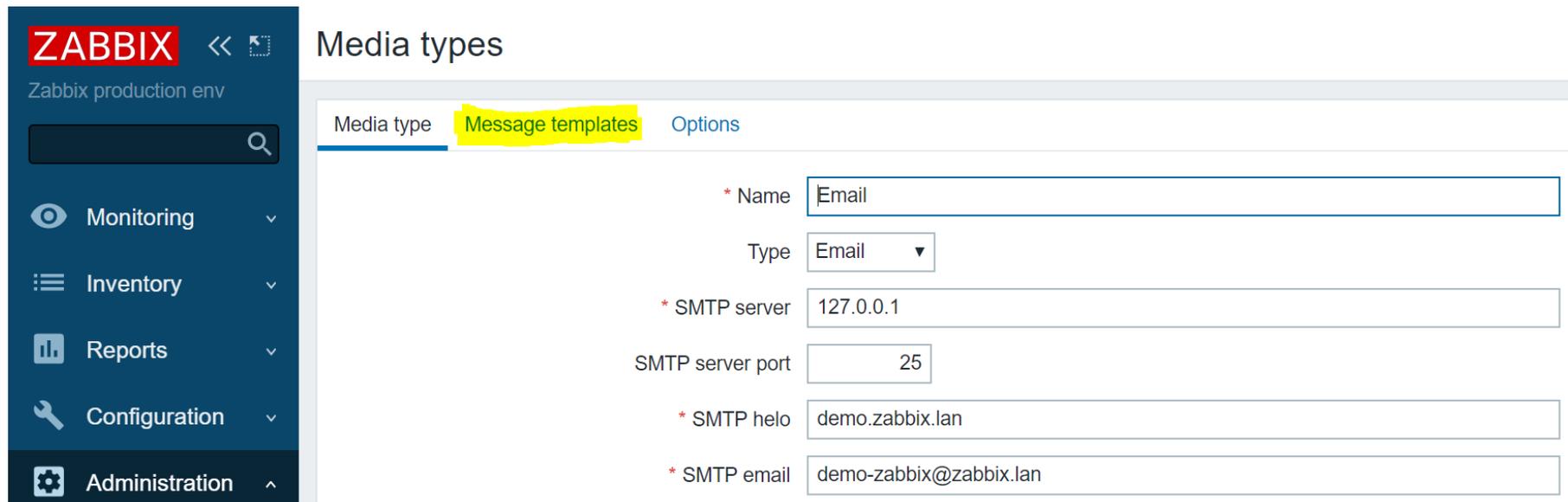
Operations

Steps	Details	Start in	Duration	Ac
<a href="#">New</a>				

\* At least one operation, recovery operation or update operation must exist.

# СООБЩЕНИЯ ПО УМОЛЧАНИЮ ДЛЯ КАЖДОГО СПОСОБА ОПОВЕЩЕНИЙ

- Всегда думайте о масштабируемости
- Могут быть пользователи с 500+ действиями
- Могут быть пользователи с 500+ способами оповещений
- Могут быть пользователи с 500+ различными стандартами для сообщений для разных ситуаций



The screenshot displays the Zabbix web interface for configuring media types. The left sidebar shows the navigation menu with 'Administration' selected. The main content area is titled 'Media types' and has three tabs: 'Media type', 'Message templates' (highlighted in yellow), and 'Options'. The 'Message templates' tab contains a form for configuring an email media type. The fields are as follows:

Field	Value
* Name	Email
Type	Email
* SMTP server	127.0.0.1
SMTP server port	25
* SMTP helo	demo.zabbix.lan
* SMTP email	demo-zabbix@zabbix.lan

# СООБЩЕНИЯ ПО УМОЛЧАНИЮ ДЛЯ КАЖДОГО СПОСОБА ОПОВЕЩЕНИЙ

- Задайте стандарт сообщений для способа оповещений
- Задайте стандарт сообщений для каждого типа событий
- Если необходимо – переопределите на уровне действий

Media type Message templates Options

Message type	Template	Actions
Problem	Problem started at {EVENT.TIME} on {EVENT.DATE} Pro...	<a href="#">Edit</a> <a href="#">Remove</a>
Problem recovery	Problem has been resolved at {EVENT.RECOVERY.TIME}...	<a href="#">Edit</a> <a href="#">Remove</a>
Problem update	{USER.FULLNAME} {EVENT.UPDATE.ACTION} problem ...	<a href="#">Edit</a> <a href="#">Remove</a>
Discovery	Discovery rule: {DISCOVERY.RULE.NAME} Device IP: {D...	<a href="#">Edit</a> <a href="#">Remove</a>
Autoregistration	Host name: {HOST.HOST} Host IP: {HOST.IP} Agent port:...	<a href="#">Edit</a> <a href="#">Remove</a>

[Add](#)

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

**Message template** ✕

Message type

Subject

Message

[Update](#) [Cancel](#)

# 05



## УЧЁТНЫЕ ДАННЫЕ SNMP НА УРОВНЕ ИНТЕРФЕЙСА УЗЛА СЕТИ

- ⊗ Избегайте ошибок при создании множества метрик
- ⊗ Проще настроить
- ⊗ SNMPv1, SNMPv2, SNMPv3 => SNMP Agent

# УЧЁТНЫЕ ДАННЫЕ SNMP НА УРОВНЕ ИНТЕРФЕЙСА УЗЛА СЕТИ

- Сложность параметров аутентификации
- Человеческий фактор – ошибки и опечатки
- Шанс ошибки возрастает вместе с увеличением числа метрик
- Одна ошибка в прототипе элементов данных влияет на хост в целом

\* Name

Type

\* Key

\* SNMP OID

Context name

Security name

Security level

Authentication protocol

Authentication passphrase

Items

All templates / Cisco 2620 SNMPv3 Applications Items 10 Triggers 3 Graphs 2 Screens Discovery rules

Item Preprocessing

\* Name

Type

\* Key

\* SNMP OID

Context name

Security name

Security level

Authentication protocol

Authentication passphrase

Privacy protocol

Privacy passphrase

Port

Type of information

\* Update interval

# УЧЁТНЫЕ ДАННЫЕ SNMP НА УРОВНЕ ИНТЕРФЕЙСА УЗЛА СЕТИ

- Вместо предложений «быть осторожнее» - минимизация возможности ошибки
- snmpv1, snmpv2, snmpv3 заменены на SNMP Agent
- Все настройки на уровне интерфейса узла сети
- Все элементы данных наследуют настройки интерфейса узла сети

* Interfaces	Type	IP address	DNS name	Connect to	Port	Default
^	SNMP	<input type="text" value="10.100.0.42"/>	<input type="text" value="net.cisco.c7600"/>	<input type="radio"/> IP <input type="radio"/> DNS	<input type="text" value="161"/>	<input checked="" type="radio"/> Remove
	* SNMP version	<input type="text" value="SNMPv2 ▼"/>				
	* SNMP community	<input type="text" value="{\${SNMP_COMMUNITY}"/>				
		<input checked="" type="checkbox"/> Use bulk requests				

[Add](#)

СПАСИБО!

